# Horizontal Federated Recommender System: A Survey

LINGYUN WANG, Zhejiang University of Technology, Hangzhou, China
HANLIN ZHOU, Zhejiang University of Technology, Hangzhou, China
YINWEI BAO, Zhejiang University of Technology, Hangzhou, China
XIAORAN YAN, Zhejiang Lab, Hangzhou, China
GUOJIANG SHEN, Zhejiang University of Technology, Hangzhou, China
XIANGJIE KONG, Zhejiang University of Technology, Hangzhou, China

Due to underlying privacy-sensitive information in user-item interaction data, the risk of privacy leakage exists in the centralized-training recommender system (RecSys). To this issue, federated learning, a privacy-oriented distributed computing paradigm, is introduced and promotes the crossing field "Federated Recommender System (FedRec)." Regarding data distribution characteristics, there are horizontal, vertical, and transfer variants, where horizontal FedRec (HFedRec) occupies a dominant position. User devices can personally participate in the horizontal federated architecture, making user-level privacy feasible. Therefore, we target the horizontal point and summarize existing works more elaborately than existing FedRec surveys. First, from the model perspective, we group them into different learning paradigms (e.g., deep learning and meta learning). Second, from the privacy perspective, privacy-preserving techniques are systematically organized (e.g., homomorphic encryption and differential privacy). Third, from the federated perspective, fundamental issues (e.g., communication and fairness) are discussed. Fourth, each perspective has detailed subcategories, and we specifically state their unique challenges with the observation of current progress. Finally, we figure out potential issues and promising directions for future research.

CCS Concepts: • **General and reference** → **Surveys and overviews;** • **Information systems** → **Recommender systems;** • **Security and privacy** → **Privacy-preserving protocols**;

Additional Key Words and Phrases: Horizontal federated learning, privacy-preserving recommender system, federated recommender system

**ACM Reference Format:**
Lingyun Wang, Hanlin Zhou, Yinwei Bao, Xiaoran Yan, Guojiang Shen, and Xiangjie Kong. 2024. Horizontal Federated Recommender System: A Survey. *ACM Comput. Surv.* 56, 9, Article 240 (May 2024), 42 pages. https://doi.org/10.1145/3656165

This work was supported in part by the National Natural Science Foundation of China under grants 62072409 and 62073295, in part by the Zhejiang Provincial Natural Science Foundation under grant LR21F020003, and in part by Zhejiang Lab under grant 2022NF0AC01.

Authors' addresses: L. Wang, H. Zhou, Y. Bao, G. Shen, and X. Kong (Corresponding author), Zhejiang University of Technology, Hangzhou, Zhejiang, China, 310023; e-mails: lywang@zjut.edu.cn, hanlinzhou@zjut.edu.cn, yinweibao@zjut.edu.cn, gjshen1975@zjut.edu.cn@zjut.edu.cn, xjkong@acm.org; X. Yan, Zhejiang Lab, Hangzhou, Zhejiang, China, 311100; e-mail: yanxr@zhejianglab.com.


ACM 0360-0300/2024/05-ART240
https://doi.org/10.1145/3656165

# 1 INTRODUCTION

## 1.1 Background

As a representative of the digital era, internet applications (e.g., online news and e-commerce) are profoundly impacting our daily lives. People can access a variety of digital resources through mobile phones, tablets, and laptops whenever and wherever. According to the latest statistics on the global digital population [151], there is a proliferation of internet users on worldwide social media platforms. While the number of users and items is simultaneously growing rapidly, companies and organizations are facing the dilemma that it is intricate to analyze each user's preference and make business decisions on personalized advertising promotion or other recommendations. Therefore, the recommender system (RecSys) is urgently required to handle the "information overload" problem.

*What Is RecSys?* RecSys can discover users' interests from historical user-item interaction data and provide reasonable recommendation lists. Effective RecSys can build a "bridge" between users and companies, which not only enables users to discover more valuable items but also serves companies with more profits. The category of RecSys is roughly divided into collaborative filtering (CF) based, content-based, and hybrid groups [215]. CF-based RecSys can calculate the similarity between items in terms of users' behaviors (known as ItemCF), or recommend favorite items of others who have behaviors similar to those of the current user (known as UserCF). Content-based RecSys leverages items' or users' auxiliary information to assist recommendation tasks. Hybrid RecSys has characteristics of both categories.

*Privacy Issues in RecSys.* The majority of RecSys train recommendation models in a single central server with extensively gathered user explicit feedback (e.g., ratings) or implicit feedback (e.g., browse, purchase, click). However, these datasets often retain privacy-sensitive information and may pose threats to users' privacy security during data collection and model construction. Although much of training data is anonymized without any additional metadata, attackers can still utilize de-anonymization approaches to mine potential user information [119].

*International Concerns for Privacy.* Additionally, people are aware of the imperative of privacy protection owning to privacy disclosure events in recent years. For example, millions of LinkedIn accounts were publicly sold on the black market, and Google+ had to shut down services because of potential privacy risks. This severe situation has promoted a series of laws and regulations: in 2018, "GDPR (General Data Protection Regulation)" came into force in the European Union, which regulates the rights of European citizens in the management of their personal data; in 2020, "CCPA (California Consumer Privacy Act)" was enacted in the United States; and in 2021, "Personal Information Protection Law" was enforced in China.

*Why Federated Recommender Systems?* Faced with the urgent challenge of guaranteeing recommendation performance and personal data security simultaneously, **Federated Learning (FL)** [111] enters researchers' vision. The critical point of FL is that users' private data are retained locally and not allowed to be shared. Moreover, the security can be further improved by integrating cryptography- or perturbation-based algorithms. It provides a new perspective for privacy-preserving RecSys and facilitates cross-field research named *Federated Recommender System (FedRec)*. The goal of FedRec is to provide users with reasonable recommendations without privacy violations. Plenty of experiments on real-world datasets have demonstrated that the federated-training mode can obtain similar or even superior recommendation performance in comparison with the centralized-training mode.

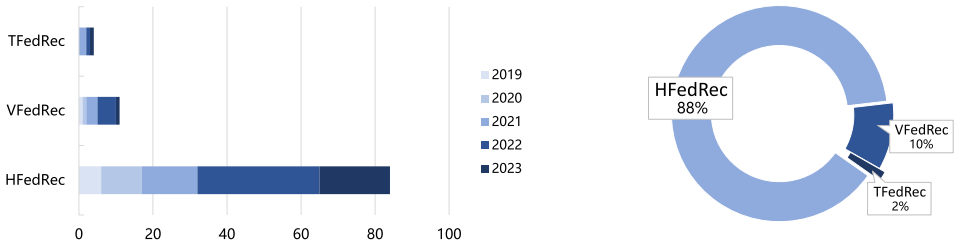
## 1.2 Motivations and Investigation

*Motivations from Related Surveys.* Yang et al. [99] first give problem definitions and research directions to *H*orizontal FedRec (HFedRec), *V*ertical FedRec (VFedRec), and *T*ransfer FedRec (TFedRec).

As a pioneering survey, the proposed taxonomy now requires further refinements to shed more light on the differences among recommendation models, privacy techniques, and federated issues. Follow-up works [4, 7, 154] fill the gap in reviewing privacy concerns and recommendation performance of FedRec to some extent. However, a more insightful survey is still needed due to the following observations:

- —From the model perspective, FedRec belonging to different learning paradigms needs to be organized coherently. For example, there are significant differences in federated procedures between deep learning and reinforcement learning (RL).
- —From the privacy perspective, past surveys do not pinpoint which object the privacy technique is applied to (e.g., raw data or gradients), especially when multiple privacy techniques are adopted in one federated framework.
- —From the federated perspective, fundamental issues, such as communication overhead and fairness, should be appreciated along with the preceding perspectives.

*Investigation into FedRec Publications.* Thanks to researchers' efforts in recent years, much worthy literature deserves attention. Through careful investigation, we sort out related works from 2019 and draw statistical charts based on the number of publications and citations in Figure 1. Both indicate that HFedRec is attracting increasing interest from researchers.




(a) Comparison of the Number of Publications (b) Comparison of the Number of Citations

Fig. 1. The historical trend of FedRec publications from 2019 to mid-2023.

*Why Do We Survey HFedRec?* There are three main reasons: First, the essence of FL has been firmly rooted in RecSys. It is complicated to classify fruitful achievements from all of HFedRec, VFedRec, and TFedRec, since different data distributions present unique challenges. Second, currently, HFedRec is the dominant research subject compared with others, as shown in Figure 1. It is intriguing how different recommendation models and privacy techniques perform in so many horizontal federations. Third, user devices can serve as federated participants in HFedRec, which makes it feasible to realize user-level privacy. Conversely, clients in VFedRec and TFedRec are limited to organizations owning massive preference data. Therefore, if our survey can depict HFedRec in depth, it will bring more significance for future research.

### 1.3 Our Contributions

According to the preceding summary, we emphasize the horizontal data distribution characteristic in this survey. Our goal is to introduce HFedRec from three different perspectives. First, from the model perspective, we divide HFedRec into statistical machine learning, deep learning, and other learning paradigms. Second, from the privacy perspective, HFedRec can incorporate cryptography, perturbation, and other privacy-preserving algorithms to enhance the privacy-preserving effect. Third, from the federated perspective, we concentrate on the fundamental issues, including

communication optimization and fairness perception. In conclusion, the main contributions of this survey are as follows:

- —We propose a hierarchical taxonomy to summarize HFedRec from model, privacy, and federated perspectives systematically. The taxonomy not only highlights current innovation in recommendation algorithms but also emphasizes the importance of privacy protection.
- —We outline the steps of federated training in different subcategories. The classical steps contain Client Selection, Broadcast, Client Computation, Aggregation, and Model Update [69]. Some steps' implementation varies with privacy-preserving and recommendation algorithms.
- —We organize existing literature into summary tables and raise unique challenges from different dimensions. Each summary table is drawn according to the corresponding perspective.
- —We point out underlying gaps and list several future research directions.

The article is divided into seven sections. Section 2 reviews the development of RecSys and compare similar terms. Section 3 formally presents our taxonomy and lists the main notations in this survey. Sections 4 through 6 introduce HFedRec from different perspectives. Section 7 presents open issues and future directions. Section 8 concludes this survey.

## 2 RELATED WORK

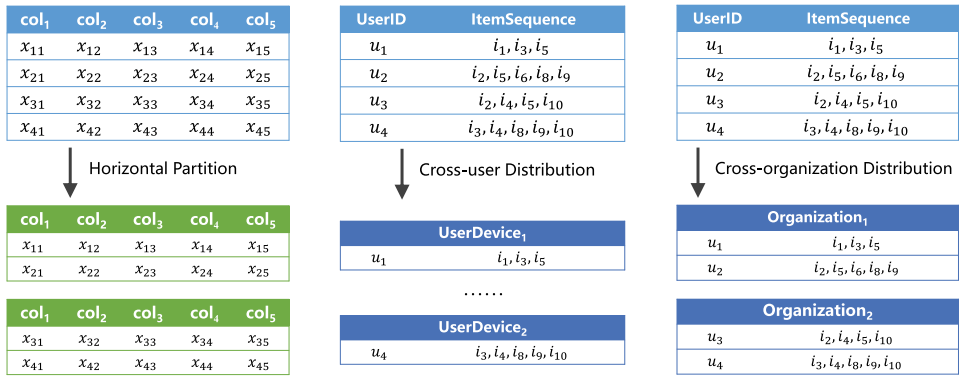
### 2.1 Overview of RecSys

RecSys assists users in selecting favorite items from massive data considering varying users' preference. In accordance with the ability to discover short- or long-term preference, RecSys can be categorized into General RecSys, Sequence-based RecSys, and Session-based RecSys. General RecSys makes use of long-term preference through static user-item interactions, assuming that users maintain steady interests over a long period. Oppositely, Sequence- and Session-based RecSys can capture dynamic short-term preference of users. The distinction is that Sequence-based RecSys utilizes the sequential dependency to model ordered historical interaction data, whereas Session-based RecSys explores the co-occurrence dependency to model ordered or unordered data with clear boundaries. Wang et al. [165] introduce the research progress of Sequence-based RecSys from traditional Markov chain to popular neural network models. More concrete definitions and comparison of Sequence- and Session-based RecSys are presented in another work by Wang et al. [164].

Apart from the preceding classification, there are various surveys to review the research progress of RecSys from different aspects. Zhang et al. [215] divide deep learning based RecSys by neural building blocks and deep hybrid models. The majority of model architectures are comprehensively introduced. Because user-item interactions can be naturally transformed into a bipartite graph, numerous studies exploit graph neural networks (GNNs) to explore potential high-order connectivity between users and items. Wu et al. [178] systematically present the development of GNN-based RecSys and provide a more sophisticated category depending on whether heterogeneous relations (e.g., knowledge graph (KG) and social network) are appendant. Moreover, POI recommendations can guide users to latent favorite locations and increase people flow in business areas or tourist attractions. Islam et al. [63] categorize POI-based RecSys in deep learning and give a full view of popular POI datasets.

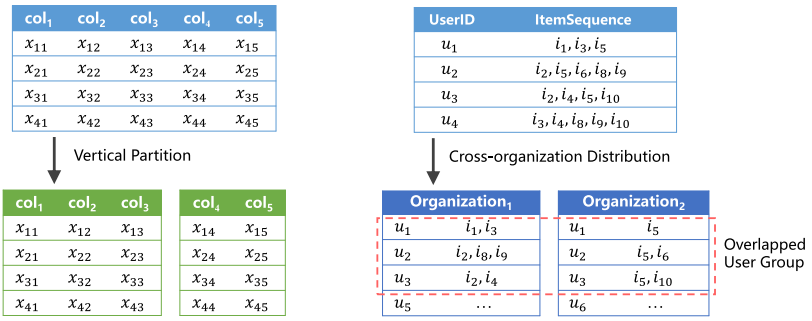
### 2.2 HFedRec vs. VFedRec

Our survey mainly groups works related to HFedRec, hence we first analyze horizontal and vertical characteristics and then give their formal definitions.




(a) Horizontal Partition of Data Table (b) Cross-user HFedRec (c) Cross-organization HFedRec

Fig. 2. An analogy of horizontal data distribution between tabular and preference data.




(a) Vertical Partition of Data Table (b) Cross-Organization VFedRec

Fig. 3. An analogy of vertical data distribution between tabular and preference data.

*How to Identify HFedRec?* Figure 2(a) illustrates the term *horizontal* with "horizontal partition" in traditional relational databases. Tabular data are split into groups by row, and rows still possess complete columns. When we place this thought into the scenario of FedRec, each row is composed of a user and interacted items (i.e., personal preference data). If an end device (e.g., mobile phone, tablet, or laptop) solely stores historical preference data for the current user, this type of FedRec is named *cross-user HFedRec* [189], as shown in Figure 2(b). However, if multiple personal preference data are gathered and stored by a platform, such FedRec is *cross-organization* (or *cross-silo*) *HFedRec* [189], as shown in Figure 2(c).

*How to Identify VFedRec?* Similarly, the term *vertical* is analogous to "vertical partition" in relational databases, where tabular data are split by columns. Based on this intuition, if organizations own overlapped user groups but different items, a vertical distribution emerges [180]. Note that before jointly training a VFedRec, organizations in Figure 3(b) need to get the intersection of users privately [18]. Besides, the federation that additional side information (e.g., user profiles and item attributes) is distributed across different clients also falls within the scope of VFedRec [42, 110].

*Summary.* Figures 2 and 3 briefly exhibit distribution differences of interaction data. Without loss of preciseness and generality, we formalize HFedRec and VFedRec as follows.

*Definition 1 (Horizontal Federated Recommender System).* Given federated clients $C$, each client retains private data $\mathcal{S}_c = \{V_c, U_c, V_c^{\text{feat}}, U_c^{\text{feat}}, R_c\}$, where $U_c$ is the user set, $V_c$ is the item set

interacted with $U_c$, $V_c^{\text{feat}}$ ($U_c^{\text{feat}}$) is item (user) features, and $R_c$ denotes local user-item interactions $(u_i, v_j, r_{ij})$. $\forall c, c' \in C, c \neq c'$, HFedRec requires $U_c \cap U_{c'} = \varnothing$ and identical feature categories to ensure the same feature space. If local user num satisfies $|U_c| = 1$, it belongs to cross-user HFedRec; otherwise, cross-organization HFedRec is formed.

*Definition 2 (Vertical Federated Recommender System).* Given federated clients $C$, each client retains private data $\mathcal{S}_c = \{V_c, U_c, V_c^{\text{feat}}, U_c^{\text{feat}}, R_c\}$, where not all variables are required. At least one client owns the interaction records $R$, and others can hold pre-aligned user profiles $\{U_c, U_c^{\text{feat}}\}$ or item attributes $\{V_c, V_c^{\text{feat}}\}$. Thus, VFedRec only adapts to the cross-organization federation.

### 2.3 Distributed RecSys vs. Federated RecSys

FedRec has a close relationship with the distributed recommender system (DRecSys), which maintains a group of computing nodes and makes full use of each node's computing resources to learn users' preference collaboratively [26, 27]. We will list their similarities and differences in this section.

*Similarities.* DRecSys and FedRec are similar in distributed computing and architecture. First, a single computing node cannot meet the requirement of scalable storage capacity on account of the increasing proportion of users and items. As datasets and tasks are decentralized among multiple computing nodes, both can effectively reduce computational overhead by parallel computation. Second, both are applicable to parameter server (PS) and peer-to-peer (P2P) architectures. PS architecture composed of a server node and client nodes can regulate globally shared model parameters; P2P architecture composed of connected client nodes can reflect higher privacy security for the eliminated server node.

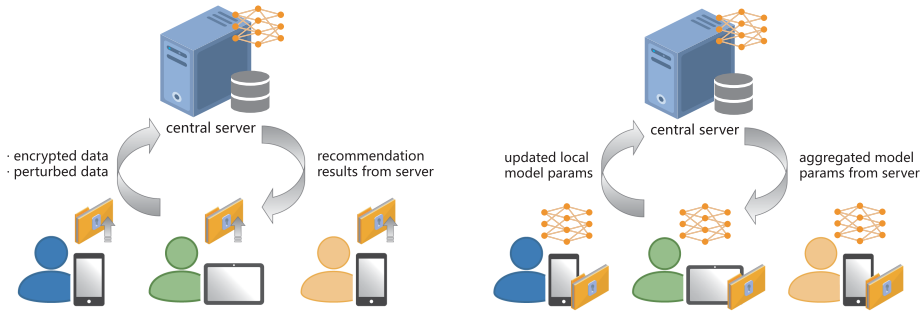
*Differences.* FedRec can be regarded as special DRecSys with the privacy constraint. Their differences consist of the following. First, FedRec pays more attention to users' privacy. Most DRecSys focus on improving distributed model accuracy and communication efficiency [47, 140], whereas FedRec gives priority to privacy protection. Second, clients/nodes play different roles. In DRecSys, clients generally refer to computing nodes in a computer cluster, and the server has absolute control over them. However, in FedRec, clients are not only computing nodes but data providers. They can regulate own data independently and decide when to quit and join the federation. Third, communication overhead is more considerable in FedRec. On account of user devices participating in the cross-user federation, their network connections may be unstable and across long-distance areas. Oppositely, computing nodes in DRecSys are often close to each other and have a stable network environment. Fourth, raw data are unbalanced among clients in FedRec. Federated clients' data are locally generated rather than uniformly allocated by the server. Data amount and distribution characteristics may vary significantly, known as the non-independent and identically distributed (IID) phenomenon [108].

*Summary.* Does a federated recommender system belong to a distributed recommender system? The answer is "Yes." While maintaining the advantages of DRecSys, FedRec faces specific challenges. Except for the communication efficiency and non-IID phenomenon, FedRec may need to trade off between privacy and model performance when privacy-preserving techniques are applied.

### 2.4 Privacy-Preserving RecSys vs. Federated RecSys

This section distinguishes another similar concept—Privacy-preserving RecSys (PRecSys) and FedRec. We will give a brief overview and then point out the implications of FL for PRecSys.

*Overview of PRecSys.* According to different privacy-preserving techniques, traditional PRecSys can be divided into Cryptography- and Obfuscation-based PRecSys. With regard to




(a) Communication Scheme of Traditional PRecSys (b) Communication Scheme of FedRec

Fig. 4. Different communication schemes between PRecSys and FedRec.

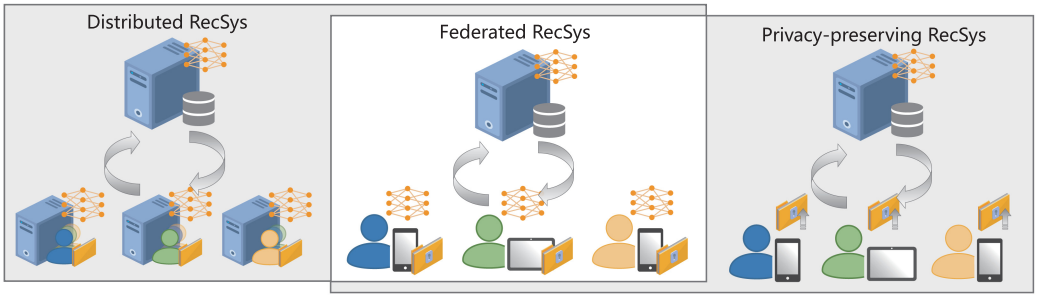



Fig. 5. Comparison of DRecSys, PRecSys, and FedRec.

Cryptography-based PRecSys, Erkin et al. [37] utilize data packing and a semi-trusted third party to develop a cryptographic protocol. Preference data are uploaded under encryption to the recommender server. Kikuchi et al. [73] exploit a quasi-homomorphic mechanism to transmit ciphertexts of user similarities between two data parties. Studies on privacy-preserving matrix factorization (MF) [75, 120, 122] explore Yao's garbled circuit (GC), oblivious transfer (OT), and homomorphic encryption (HE) to encrypt item-rating pairs from users. With regard to Obfuscation-based PRecSys, Polat and Du [130] allow the server to collect private preference data from users with a randomized perturbation scheme. McSherry and Mironov [113] apply centralized differential privacy (DP) to RecSys with more rigorous theoretical guarantee. Kikuchi and Mochizuki [74] utilize randomized response (RR) to disguise local preference data before being submitted to the server. The authors devise a posterior probability distribution function to reconstruct the original distribution from disguised data.

*The Role of FL in PRecSys.* Whereas the preceding studies obfuscate or encrypt raw preference data, private data leave local devices and are exposed in public. The proposal of FL [111] provides a new idea for PRecSys. In FedRec, only intermediate results of model training are communicated between the server and clients, which effectively avoids privacy risks arising from centralized storage of user behavior data. Therefore, for the cross-user distribution, the main difference between FedRec and PRecSys lies in the communication scheme, as shown in Figure 4. Note that past cryptography- and obfuscation-based approaches are still necessary to ensure the security of FedRec.

*Summary.* Does a federated recommender system belong to a privacy-preserving recommender system? The answer is also "Yes." FedRec is a further improvement on traditional PRecSys. Moreover, we draw Figure 5 to describe the relationship among FedRec, DRecSys, and PRecSys. As a

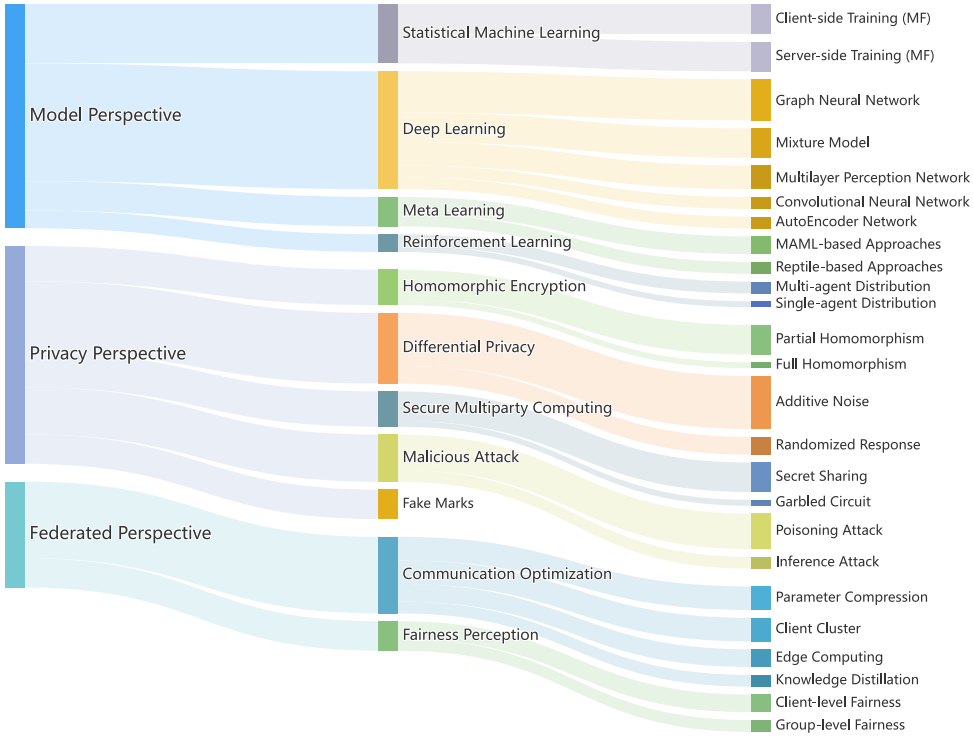



Fig. 6. Proposed taxonomy of HFedRec.

dominant component of FedRec, HFedRec is promising to digest both advantages of DRecSys and PRecSys.

## 3 A TAXONOMY OF HFEDREC

It is incomplete to summarize fruitful research achievements only from recommendation models, privacy-preserving algorithms, or federated issues. Either of three aspects requires comprehensive summaries (Figure 6 illustrates our taxonomy of HFedRec):

— *HFedRec from the model perspective*: This perspective outlines recommendation studies from statistical machine learning, deep learning, meta learning, and RL. For example, in machine learning approaches, most of works leverage MF to formulate HFedRec, and thus we cover training details from client and server sides. Due to the special research questions posed by federated graph learning, we have devoted a more detailed comparison to highlight the differences among GNN-based HFedRec.
— *HFedRec from the privacy perspective*: This perspective partitions five subcategories in terms of HE, DP, secure multi-party computing (SMC), malicious attack, and fake marks. We take into account more specific encryption and obfuscation techniques, such as RR and secret sharing (SS).
— *HFedRec from the federated perspective*: This perspective includes communication optimization and fairness perception, of which each is the fundamental issue raised by FL. We also present the preliminary investigation of HFedRec regarding federated scalability and personalized federation in Section 7 w.r.t. future directions.

Table 1. Main Notations

| Notations | Description | Notations | Description |
|---|---|---|---|
| $U$, $V$, $C$ | user set/item set/client set | $\boldsymbol{v}'_j$ | embedding of $v_j$ not interacted with $u_i$ |
| $u_i$, $v_j$ | user $i$ in set $U$/item $j$ in set $V$ | $r'_{ij}$ | virtual rating of $u_i$ to $v_j$ |
| $N_U$, $N_V$, $N_C$ | user num/item num/client num | $U_{v_j}$ | user set interacted with $v_j$ |
| $\theta$ | model parameter | $U'_{v_j}$ | user set not interacted with $v_j$, $U'_{v_j} = U \backslash U_{v_j}$ |
| $d_u$, $d_v$ | dimension of $\boldsymbol{u}_i$, $\boldsymbol{v}_j$ | $I_{ij}$ | indicator variable, $I_{ij} \in \{0, 1\}$ |
| $\boldsymbol{u}_i$ | embedding of $u_i$, $\boldsymbol{u}_i \in \mathbb{R}^{d_u}$ | $\nabla_{\boldsymbol{u}_i}$, $\nabla_{\boldsymbol{v}_j}$ | gradients of $\boldsymbol{u}_i$, $\boldsymbol{v}_j$ |
| $\boldsymbol{U}$ | user embedding matrix, $\boldsymbol{U} \in \mathbb{R}^{N_U \times d_u}$ | $\lambda$, $\eta$ | regularization parameter/learning rate |
| $\boldsymbol{v}_j$ | embedding of $v_j$, $\boldsymbol{v}_j \in \mathbb{R}^{d_v}$ | $t$, $r$ | $t$-th local epoch/$r$-th global communication round |
| $\boldsymbol{V}$ | item embedding matrix, $\boldsymbol{V} \in \mathbb{R}^{N_V \times d_v}$ | $N_S$, $N_T$ | num of total training/testing samples |
| $r_{ij}$, $\hat{r}_{ij}$ | real/predicted rating of $u_i$ to $v_j$ | $n_s^i$, $n_t^i$ | num of training/testing samples of $u_i$ |
| $V_{u_i}$ | item set interacted with $u_i$ | $\mathrm{E}(\cdot)$, $\mathrm{D}(\cdot)$ | encryption/decryption algorithm |
| $V'_{u_i}$ | item set not interacted with $u_i$, $V'_{u_i} = V \backslash V_{u_i}$ | $L(\cdot)$, $l(\cdot)$ | loss function |

We will give more detailed classifications in Sections 4 through 6 and sort out related work into summary tables respectively. Table 1 lists the main notations throughout the article.

## 4 HFEDREC FROM THE MODEL PERSPECTIVE

We put representative works from the model perspective in Table 2, including learning paradigm, recommendation models, recommendation problems, federated distribution types, and dataset types. We confirm dataset types by empirical studies in their own papers. It is difficult to reflect the uniqueness of some publications in Table 2, such as $\text{FedRec}_{\text{L}}$ [92], FedRec++ [91], PerFedRec [105], and PerFedRec++ [106]. More details can be seen in the following sections and Tables 3 through 6.

### 4.1 Statistical Machine Learning Approaches for HFedRec

According to Table 2, MF is the dominant approach in machine learning based HFedRec, aiming to decompose user embeddings $u_i \in \mathbb{R}^{d_u}$ and item embeddings $v_j \in \mathbb{R}^{d_v}$ from the sparse collaborative matrix $R \in \mathbb{R}^{N_U \times N_V}$. Under the parameter server architecture, the typical federated training steps of MF are listed as follows:

— *Step 1*: The server initializes global item embedding matrix $\boldsymbol{V} \in \mathbb{R}^{N_V \times d_v}$ and samples a batch of clients $C$ to start the federated training process.
— *Step 2*: Selected clients $C$ download the latest $\boldsymbol{V}$ from the server, then utilize local implicit or explicit feedback to update $\boldsymbol{u}_i$ and compute $\nabla_{\boldsymbol{v}_j}$.
— *Step 3*: After clients complete local training in parallel, only item gradients are transmitted to the server.
— *Step 4*: The server aggregates gradients uploaded by clients and updates $\boldsymbol{V}$. Then the server distributes the latest $\boldsymbol{V}$ to clients for the next communication round.
— *Step 5*: Repeat steps 2 through 4 until convergence.

Accordingly, involved data can be divided into public parts and private parts. *Public parts* mainly contain the global item embedding matrix $\boldsymbol{V}$ and gradients $\nabla_{\boldsymbol{v}}$ (client ID, round ID, and other auxiliary information [81] are omitted), which can be shared and transmitted between federated participants. *Private parts* are user embeddings and original user-item interaction data preserved in local devices. We will describe details from client and server sides in the following subsections.

Table 2. Representative Works in HFedRec from the Model Perspective

| Publication | Type | Model | Prob | U/O | Datasets | Publication | Type | Model | Prob | U/O | Datasets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FCF [5] | M | WMF | G1 | U | Movie | JointRec [33] | D | CNN | G1 | U | Movie |
| FedRec$_L$ [1] [92] | M | PMF | G1 | U | Movie | FedPOIRec [128] | D | CNN | S2 | U | Check-in |
| FedRec++ [91] | M | PMF | G1 | U | Movie | FedGME [127] | D | AE | G1 | O | Stock |
| DFedRec [89] | M | PMF | G1 | U | Movie | ReFRS [61] | D | AE | S1 | U | Movie, Music |
| FedMF [13] | M | PMF | G1 | U | Movie | CPF-POI [192] | D | MM | S2 | U | Check-in |
| RFPG [31] | M | BPRMF | S2 | U | Check-in | PrefFedPOI [217] | D,R | MM | S2 | U | Check-in |
| DFSR [103] | M | PMF | G1 | U | Movie | FedNewsRec [133] | D | MM | G1 | U | News |
| LightFR [208] | M | MF-L2H | G1 | U | Movie, Product | Efficient-FedRec [193] | D | MM | G1 | U | News |
| FCF-BTS [72] | R | MAB | G1 | U | Movie, Music, News | KG-FedTrans4Rec [171] | D | MM | S3 | O | Movie, Music, Book |
| FPAE [101] | R | MAB | G1 | O | Product | FedSeqRec [80] | D | MM | S1 | O | Movie, Video |
| Fed2-UCB [147] | R | MAB | G1 | U | Movie | Limited Negatives [123] | D | MLP | G1 | O | Movie |
| FedPerGNN [174] | D | GNN | G3 | U | Movie, Music | HPFL [177] | D | MLP | G1 | O | Movie, Edu |
| PerFedRec [105] | D | GNN | G3 | U | Movie, Book, Business | PFedRec [205] | D | MLP | G1 | U | Music, Video |
| PerFedRec++ [106] | D | GNN | G3 | U | Book, Check-in, Business | FedFast [117] | D | MLP | G1 | U | Movie, Product, Business |
| GPFedRec [204] | D | GNN | G3 | U | Movie, Music | FedMeta [17] | T | MAML | G1 | U | Mobile Service |
| FedHGNN [183] | D | GNN | G3 | U | Cite, Business | MetaMF [93] | T | MAML | G1 | U | Movie, Product |
| DGREC [227] | D | GNN | G3 | U | Movie, Book, Check-in | Fed4Rec [221] | T | MAML | G1 | U | News |
| SemiDFEGL [135] | D | GNN | G2 | U | Movie, Check-in, Business | FedIS [30] | T | Reptile | G1 | U | Movie, Music, Check-in, Game |
| CdFed [214] | D | GNN | G2 | U | Product | DP-PrivRec [163] | T | Reptile | G1 | U | Movie, APP |

**Type:** M: Statistical Machine Learning, D: Deep Learning, T: Meta Learning, R: Reinforcement Learning.
**U/O:** U: Cross-User Federation, O: Cross-Organization Federation.
**Model:** PMF: Probabilistic Matrix Factorization [143], WMF: Weighted MF [57], BPRMF: Bayesian Personalized Ranking based MF [137], MAB: Multi-Armed Bandit, GNN: Graph Neural Network, MLP: Multi-Layer Perceptron, CNN: Convolutional Neural Network, MM: Mixture Model, AE: Autoencoder, MAML: Model-Agnostic Meta Learning [41], Reptile: Reptile [121], MF-L2H: MF based on Learning to Hash [161].
**Prob (recommendation problem):** Referring to Table 7 in Appendix A for meanings of index codes.

[1]To avoid confusion with the abbreviations in this survey, the character "L" is added at the lower-right corner.

*4.1.1 Client-side Training.* Most MF-based HFedRec are the federation of probabilistic matrix factorization (PMF) [143] or other variants [57, 149]. PMF formulates the loss function as

$$L = \frac{1}{2}\sum_{i=1}^{N_U}\sum_{j=1}^{N_V} I_{ij}\left(r_{ij} - \boldsymbol{u}_i\boldsymbol{v}_j\right)^2 + \frac{\lambda_U}{2}\sum_{i=1}^{N_U}\|\boldsymbol{u}_i\|_2^2 + \frac{\lambda_V}{2}\sum_{j=1}^{N_V}\|\boldsymbol{v}_j\|_2^2, \tag{1}$$

where $I_{ij} = 1$ when $u_i$ rated $v_j$ and $I_{ij} = 0$ otherwise. $\lambda_U, \lambda_V$ are regularization parameters of $U, V$, respectively. When updating $\boldsymbol{u}_i$ and $\boldsymbol{v}_j$ with gradient descent, we can derive the partial derivative:

$$\frac{\partial L}{\partial \boldsymbol{u}_i} = -\sum_{j=1}^{N_V} I_{ij}(r_{ij} - \boldsymbol{u}_i\boldsymbol{v}_j)\boldsymbol{v}_j + \lambda_U\boldsymbol{u}_i, \quad \frac{\partial L}{\partial \boldsymbol{v}_j} = -\sum_{i=1}^{N_U} I_{ij}(r_{ij} - \boldsymbol{u}_i\boldsymbol{v}_j)\boldsymbol{u}_i + \lambda_V\boldsymbol{v}_j. \tag{2}$$

Owing to privacy-sensitive user embeddings required to be maintained locally [13], it is impossible to leverage all user embeddings to update item embeddings by $\boldsymbol{v}_j = \boldsymbol{v}_j - \eta\partial L/\partial \boldsymbol{v}_j$. For handling this issue, Ammad-Ud-Din et al. [5] first propose federated collaborative filtering (FCF) and define the basic training steps mentioned previously. Only the current user embedding is utilized to calculate local item gradients. Most of studies [13, 32, 34, 89, 115, 188] follow the solution

in FCF. Generally, $\nabla_{\boldsymbol{u}_i^t}$ and $\nabla_{\boldsymbol{v}_j^t}$ at the $t$-th local epoch are computed as follows:

$$\nabla_{\boldsymbol{u}_i^t} = -\sum_{j \in V_{u_i}} \left(r_{ij} - \hat{r}_{ij}\right) \boldsymbol{v}_j^{t-1} + \lambda \boldsymbol{u}_i^{t-1}, \quad \nabla_{\boldsymbol{v}_j^t} = -\left(r_{ij} - \hat{r}_{ij}\right) \boldsymbol{u}_i^{t-1} + \lambda \boldsymbol{v}_j^{t-1}. \tag{3}$$

Especially, FedRec$_L$ [92] and FedRec++ [91] randomly sample unrated items from $V_u^{'}$ and assign them virtual ratings (detailed in Section 5.4 w.r.t. fake marks). Instead of an entire item set, clients upload interacted item gradients with fake entries to the server. FCMF [185] explores the scenario that organization-partitioned preference data are heterogeneous (e.g., scoring ranges of $\{1, 1.5, \ldots, 4.5, 5\}$ and $\{1, 2, \ldots, 9, 10\}$). The local interaction data of two parties are treated as a target matrix and an auxiliary matrix, where the latter is served as complementary information to the former. DFSR [103] joins a contrastive self-supervised task to the target recommendation task. The contrastive goal is to close the distance between positive embeddings of local user and social friends and local items and global items. In a decentralized setting, DFedRec [89] performs user-user collaboration depending on a social network constructed by a temporary server. According to social links, clients can exchange item gradients with their neighbors who have co-interacted items.

*4.1.2 Sever-side Training.* At the $r$-th communication round, clients compute local item gradients in parallel and submit gradients to the server. Before updating the global item embedding matrix $\boldsymbol{V}$, the server needs to aggregate uploaded gradients:

$$\boldsymbol{v}_j^r = \boldsymbol{v}_j^{r-1} - \eta \nabla_{\boldsymbol{v}_j^r} = \boldsymbol{v}_j^{r-1} - \eta \sum_{i=1}^{N_C} \nabla_{\boldsymbol{v}_{ij}^r}. \tag{4}$$

For eliminating the precision loss caused by pseudo interacted items, FedRec++ [91] selects some denoising clients to aggregate gradients $\nabla_{\boldsymbol{v}_j^r}$ from $U_{v_j}^{'}$. Differently, DFedRec [89] selects a temporary server to receive real and pseudo interacted items from other clients. Based on co-interacted items, the server constructs a social network containing user nodes and social links to guide decentralized collaboration, which is analogous to the global graph reconstruction in Section 4.2.3 w.r.t. GNN-based HFedRec. In addition, FCF [5] and A-RFRS [16] analyze the Adam optimizer applied to MF. The server first aggregates received moments from clients, then adaptive learning rate and global item embeddings are updated by aggregated moments.

*4.1.3 Summary and Challenges.* Recent studies have explored federated MF with various privacy-preserving techniques, such as HE [13, 185], DP [115, 185], and SMC [34, 94] but have paid little attention to MF itself. The first challenge is that *MF encounters a dilemma in fusing side information*. To this issue, based on users' active regions, RFPG [31] privately incorporates geographical locations into MF to enhance POI recommendation performance. Second, *sparse and large-scale user-item interactions account for the efficiency bottleneck*, especially in the resource-constrained federation. LightFR [208] is the first effort to highlight the factorization ability through learning to hash [161]. Users and items are represented as compact binary codes to compute user-item similarity efficiently in a discrete Hamming space. More research is encouraged to break through vanilla machine learning algorithms in HFedRec.

## 4.2 Deep Learning Approaches for HFedRec

In contrast to statistical machine learning approaches, deep learning based RecSys can effectively mine deeper user-item relationships with powerful non-linear transformation ability [28]. The federated training steps of deep learning based HFedRec can be summed up as follows:

— *Step 1*: The server initializes weights of the global model and samples a batch of clients to start the federated training steps.
— *Step 2*: Selected clients download weights of the global model from the server, then utilize local implicit or explicit feedback to update local models.
— *Step 3*: After clients complete local training in parallel, gradients or weights are transmitted to the server.
— *Step 4*: The server aggregates params uploaded by clients and updates the global model. Then the server distributes latest weights to clients for the next communication round.
— *Step 5*: Repeat steps 2 through 4 until convergence.

Apart from item embeddings, whether hidden layers' weights need aggregating also remains a decision. The server can solely take charge of aggregating uploaded parameters without maintaining a global model. Then, we categorize HFedRec in terms of different types of neural building blocks.

*4.2.1 Multi-layer Perceptron Network.* The multi-layer perceptron network (MLP) is a concise neural network and basic module for other complex models [132]. Kalloori and Klingler [70] choose neural collaborative filtering (NCF) [50], a generalized MF based on MLP, to model user-item interactions among organizations. Further studies of federated NCF focus on communication efficiency [117] and privacy security [65, 129]. To customize local models, PFedRec [205] fine-tunes global item embeddings and learns the user-specific MLP for the final scoring. Ning et al. [123] explore spreadout regularizer [218] and batch-insensitive losses to build a dual-tower recommendation model when sampling negatives from non-IID data accounts for the performance degradation. HPFL [177] devises a hierarchical local model. Its private component embeds user/item features which are privacy sensitive, then a public component fuses open item knowledge into private embeddings as desensitization.

*4.2.2 Convolutional Neural Network.* The convolutional neural network (CNN) composed of convolutional layers, pooling layers, and fully connected layers is prevalent in processing images, videos, and natural languages [90]. JointRec [33] combines CNN with word embedding to model user and video representations from local textual information (e.g., user profiles and video description in a document). Then, convoluted features are fed into PMF for predicting users' ratings on videos. Inspired by CASER [157], FedPOIRec [128] attempts to mine sequential patterns of user behaviors. Different from images and texts, convolutional kernels with horizontal and vertical shapes are applied to the embedding matrix of short-term interacted items.

*4.2.3 Graph Neural Network.* GNN is a powerful backbone for modeling graph structure data in manifold domains [100, 181]. When GNN encounters FL, especially the cross-user federation, each client only holds an ego-network with the current user node and interacted item nodes, which confines graph signals' propagation over high-order paths. To capture the high-order connectivity among subgraphs distributed across clients, how to devise a **local graph expansion (LGE)** strategy is a prominent research question. Considering LGE and graph heterogeneity, Table 3 further compares existing works of GNN-based HFedRec.

FGC [195] and FL-GMT [87] are defined in the cross-organization federation which naturally mitigates the constraints imposed by over-sparse local graphs. Some [87, 102, 107] assume that social links has been locally available, wherein FL-GMT [87] and FeSoG [102] design unique attention mechanisms to weigh the importance of user-item and user-user pairs. For the alleviation of local computation pressure, FedGR [107] assigns a user model to each client and an item model to the server, respectively. In privacy-sensitive scenarios of Industrial IoT, FGC [195] recommends suitable IoT services for local devices by a federated graph convolutional network.

Table 3. Comparison of GNN-Based HFedRec from Graph Heterogeneity and LGE

| Publication | U/O | HN | HL | LGE | Model | Publication | U/O | HN | HL | LGE | Model |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FedPerGNN [174] | U | – | Social | Search | Frame | FedGR [107] | U | – | Social | – | Att |
| PerFedRec [105] | U | – | Social | Search | Frame | GPFedRec [204] | U | – | Social | Global | Frame |
| PerFedRec++ [106] | U | – | Social | Search | Frame | FedHGNN [183] | U | Multi | Multi | Global | Att |
| FGC [195] | O | – | – | – | Conv | CdFed [214] | U | – | – | Cluster | Frame |
| FL-GMT [87] | O | – | Social | – | Att | SemiDFEGL [135] | U | – | – | Cluster, P2P | Frame |
| FeSoG [102] | U | – | Social | – | Att | DGREC [227] | U | – | Social, Hyper | Global, P2P | Conv, Att |

**U/O:** U: Cross-User Federation, O: Cross-Organization Federation.
**HN (heterogeneous nodes except users and items):** Multi: Multi-type nodes.
**HL (heterogeneous links):** Multi: Multi-type links; Social: Social links between users;
Social: Social links between users; Hyper: Heterogeneous hyperedge connecting a subset of nodes [6].
**LGE (Local Graph Expansion):** Search: Search for neighbors based on matching mutual information; Cluster: Cluster of similar clients; Global: Global graph reconstruction; P2P: Peer-to-peer collaboration between clients.
**Model:** Frame: GNN-Agnostic Framework; Att: Graph Attention Network (GAT); Conv: Graph Convolutional Network (GCN).

Without the assumption of prior social knowledge, FedPerGNN [174] introduces a third party to search similar users with co-interacted items, whereas PerFedRec [105] and PerFedRec++ [106] follow its practice to enrich local graphs with neighbors. Compared with PerFedRec, PerFedRec++ regards client sampling and noise injection from privacy-preserving techniques as data augmentation. A contrastive self-supervised task is adopted to pre-train models. Luo et al. [105, 106] employ a clustering method to improve communication efficiency detailed in Section 6.1. Another idea is to privately reconstruct a pseudo global graph connecting data islands. GPFedRec [204] reconstructs a social network in the server through user-user relevance computed by uploaded item embeddings. The server-side GNN then aggregates social-guided item embeddings to compensate for user-user collaborative signals. FedHGNN [183] performs meta path based recommendation and generalizes graph heterogeneity to multi-type nodes and links. Given public knowledge (e.g., item types) and perturbed local interactions, the server reconstructs a global heterogeneous information network (HIN) only once and then sends corresponding meta paths back to clients. In a decentralized federated setting, DGREC [227] builds an item hypergraph with tag information and performs an interest pooling operation to model user preference. Clients reconstruct social networks referring to the work of Krasanakis et al. [78] and share update gradients guided by social links. Moreover, SemiDFEGL [135] decides which neighbors a client can communicate with through a clustering method. According to uploaded local ego-network representations, the server will cluster clients into groups, and P2P parameter passing is also constrained at the graph level. Similarly, after the server clustering clients by uploaded model weights, CdFed [214] allows clients to pass update gradients to intra-cluster neighbors.

*4.2.4 Autoencoder Network.* The autoencoder network is an unsupervised learning model. An encoder maps the input data into a feature vector $z$, and a decoder attempts to reconstruct input data according to $z$. Then, $z$ is regarded as a salient feature for subsequent recommendation tasks [207]. ReFRS [61] employs two variational autoencoders (VAEs) [145] in both server and client sides for sequential recommendation. To avoid high variance and posterior collapse, the client-side VAE discretizes temporally encoded item sequences through a vector quantization technique. The server-side VAE takes uploaded encoder weights as input and models their semantic similarity. Within-cluster encoders are aggregated after $k$-means clustering. FedGME [127] defines concept-stock recommendation problem on public stock reports, private forum comments, and experts' knowledge. Transformer graph layers [148] act as an encoder to learn fused embeddings of concepts and stocks, whereas linear layers aim to reconstruct the two source embeddings.

*4.2.5 Mixture Model.* The mixture model refers to the network composed of multi-type neural building blocks. FedNewsRec [133] aims at privacy-preserving news recommendations, comprising two core modules: (i) the user module is a self-attention-extended LSTUR model [43] and learns user embeddings from historical user behaviors, and (ii) the news module learns news embeddings from news titles and embodies word embedding, attention, and convolutional layers. Efficient-FedRec [193] further puts the heavyweight news model on the server side and the lightweight user model on the client side to offload clients' computational overhead. To learn preference evolution with subtle temporal dependency, FedSeqRec [80] combines low-rank tensor projection and gated recurrent unit (GRU) layers to model long- and short-term interests, respectively. CPF-POI [192] first locally removes location noises within historical trajectories. Then embedded location and timestamp records are attentively fused and fed into GRU layers considering the spatial-temporal interval. Similarly affected by the sparsity of trajectories, PrefFedPOI [217] attentively fuses historical recent and periodic preference with the awareness of time interval. Relying on the interpretable inference and structural semantics of KGs, KG-FedTrans4Rec [171] injects item knowledge distilled by GCN into Transformer-based sequential recommendation.

*4.2.6 Summary and Challenges.* Despite validated on specific models, some works [105, 128, 195] tend to be generic frameworks suitable for a broader range of neural networks. Meanwhile, some networks [102, 127, 192] devised for the horizontal federation can also work well in the centralized setting. However, computation overhead for high-dimensional model weights and embedding matrices are more expensive than statistical machine learning approaches. One specific challenge is *how to design federated models that concentrate intensive computing on the server side.* Efficient-FedRec [193] and FedGR [107] assign a burdensome item representation task to the server, which shares an idea similar to split learning [158]. Knowledge distillation [88] also enables clients to own tinier models, introduced in Section 6.1. For sequential and graphical data, another challenge is *how to enhance feature diffusion across local sequences and graphs.* A vanilla federated averaging strategy [111] is insufficient to explicitly capture common sequential and graphical patterns among clients. More federated models are expected to highlight the computation pressure and data characteristics rather than copy the federated procedure directly.

### 4.3 Meta Learning Approaches for HFedRec

Machine learning and deep learning based HFedRec have combined FL with many kinds of recommendation algorithms, but there are still two issues:

(1) Different users tend to present personalized behavior patterns especially in cross-user HFedRec. For instance, some have a long-term preference for digital products, whereas others prefer daily necessities according to their short-term needs.

(2) All clients' recommendation tasks are treated as single learning problem to train a global model performing well for most of them. The global model may ignore uncommon preference.

Therefore, meta learning comes into the field of HFedRec. The nucleus concept of meta learning is "learning to learn," and the emphatic problem is "how to learn a model faster and better in the near future." From the view of task distribution, each client's user-item interaction pattern can be regarded as a training *task*. Meta learning intends to obtain a general-purpose learning algorithm that quickly adapts to a new scenario within a few local training steps [54]. In the following subsections, we will describe two typical meta algorithms of model-agnostic meta learning (MAML) [41] and Reptile [121] explored in HFedRec.

*4.3.1 MAML-based Approaches.* In the case of MAML, local preference data must be divided into support set and query set. The support set first updates model parameters $\theta_0$, then the query

set points to the real gradient descent direction:

$$\begin{aligned}\theta_i &= \theta_0 - \eta_1 \nabla_\theta L_{D_i^s}(f_{\theta_0}) \\ \theta_0 &= \theta_0 - \eta_2 \frac{1}{|T|} \sum_{T_i} \nabla_\theta L_{D_i^q}(f_{\theta_i}),\end{aligned} \tag{5}$$

where $D_i^s$ and $D_i^q$ are the support set and query set of the $i$-th task $T_i$, $\eta_1$ and $\eta_2$ are learning rates at different training stages, and $\theta_i$ denotes the updated model parameters by support set $D_i^s$. When MAML meets FL, clients serve as *tasks*, and the number of tasks $|T|$ also refers to the number of sampled clients. The summation of gradients $\nabla_\theta L_{D_i^q}(f_{\theta_i})$ is computed in the server. FedMeta [17] first defines the primary procedure:

- *Step 1*: The server initializes model parameters $\theta_0$ and samples a batch of clients $C$ (equivalent to tasks $\{T_i, i \in [C]\}$) to start the federated training process.
- *Step 2*: Selected clients $C$ download model initialization parameters $\theta_0$ from the server, then utilize the support set $D_i^s$ to train local models by one or more gradient descent steps.
- *Step 3*: After completing local training steps in parallel, clients send test gradients computed by the query set $D_i^q$ to the server.
- *Step 4*: The server aggregates test gradients uploaded by clients and updates model initialization parameters $\theta_0$. Then the server distributes initialization parameters to clients for the next communication round.
- *Step 5*: Repeat steps 2 through 4 until convergence.

Besides, Fed4Rec [221] combines first-order MAML with bi-directional GRUs and attention layers. It considers two different user groups participating in HFedRec: public users who agree to share personal data and private users who disagree to share. MetaMF [93] is a meta MF that consists of a collaborative memory (CM) module, a meta recommender (MR) module, and a rating prediction (RP) module. The RP module is deployed in local devices for predicting ratings of items. CM and MR modules, deployed in the server, aim to encode users' behaviors and generate parameters of private RP modules, respectively.

*4.3.2 Reptile-based Approaches.* Compared with MAML, Reptile [121] consumes less computational resources without twice derivative calculations and separation of the support set and query set. The key distinction consists of the update of model initialization parameters:

$$\theta_0 = \theta_0 + \eta \frac{1}{|T|} \sum_{T_i} (\theta_i - \theta_0). \tag{6}$$

Reptile directs model initialization parameters to the final updated parameters learned from tasks. DP-PrivRec [163] implements the meta update by Reptile to quickly adapt inactive users to local preference patterns. Using Taylor series expansion to approximate the Reptile-style aggregation in the server, FedIS [30] additionally gives a theoretical proof that the meta update has the ability of global model generalization.

*4.3.3 Summary and Challenges.* Meta learning does not require a global model to reach optimal performance of all clients at once, and local models can converge within fewer training epochs, of which both make it possible to build efficient HFedRec. Therefore, a latent challenge is *how to further combine efficiency-enhancing approaches with the meta paradigm.* For instance, LightFR [208] and EfficientFedRec [193] mentioned in Section 4.1 and 4.2 have made efficiency improvement in their own way. It remains a surprise when meta learning combines with existing efforts. Furthermore, since federated meta learning aims at a fast adaption from the global model

to local heterogeneous tasks, it is natural to promote another subject named **personalized federated learning (PFL)** [155]. One advantage of PFL is to handle non-IID data varying in size and distribution. Thus, another significant challenge is *how to incorporate personalization strategies in PFL*. We also include this point as a future direction and outline recently preliminary investigation of PFL-based HFedRec in Section 7.6.

### 4.4 Reinforcement Learning Approaches for HFedRec

RL is dedicated to the consistent interaction process between the agent and environment rather than learning pre-prepared tasks regarding diverse scenarios [125]. In every agent-environment interaction, an agent determines the next action according to the observed state of the environment, wherein the multi-armed bandit (MAB) is a classic RL problem:

(1) The *agent* is a player (i.e., RecSys), and the *environment* is a bandit (i.e., user) with $K$ arms (i.e., items). At each time slot, the player decides which arm to pull from the set of $K$ arms.

(2) Each pulled arm has different probability distribution for gaining coins (i.e., rewards), while the player aims to know which arm can maximize the total reward during the playing process.

Therefore, RL-based RecSys can give reasonable recommendations in dynamic environments, regardless of whether the current scenario is present in datasets or not. We classify current RL-based HFedRec into multi-agent and single-agent distributions to detail their solutions.

*4.4.1 Multi-agent Distribution.* The horizontal federation places each client in an independent *environment*, and clients cannot interfere with each other directly. Thus, local models can be regarded as individual *agents* and play the same set of $K$ arms. The server takes charge of aggregating observations uploaded by clients. Fed2-UCB [147] first defines a systematic federated MAB framework experimented on recommendation tasks, and the generic steps are as follows:

- *Step 1*: All clients initialize active arms $K_{\text{act}}$ (i.e., candidate items), and the server samples a batch of clients $C$ to start federated training steps.
- *Step 2*: Selected clients pull each active arm for constant times. Sample means $\{\overline{\mu}_{c,k} | c \in C, k \in K_{\text{act}}\}$ are updated based on users' feedback and transmitted to the server. *(Sub-Round I.)*
- *Step 3*: The server updates the global $\sigma$-sub-Gaussian distribution with aggregated sample means $\sum_{c \in C} \overline{\mu}_{c,k}/|C|$. Then, double upper confidence bounds (UCBs) [8] are utilized to determine the confidence bound and suboptimal arms $K_{\text{eli}}$ to be eliminated.
- *Step 4*: Selected clients download the elimination arm set from the server to decrease active arms $K_{\text{act}} = K_{\text{act}} \backslash K_{\text{eli}}$. *(Sub-Round II.)*
- *Step 5*: Repeat steps 2 through 4 until one active arm is left.

The double UCBs aim to handle uncertainty brought by client sampling and arm sampling. The former means that the global bandit model may not fairly select the optimal arm due to the limitation of incorporating only part clients in one communication round, whereas the latter is common in the Exploration-Exploitation dilemma of RL [210]. Moreover, Fed2-UCB has a simplified version named Fed1-UCB in a special case where the client sampling is excluded, and all clients join in the decision-making process. Similarly, FPAE [101] adapts federated MAB to aspect-based recommendations where users can rate different aspects of an item, such as price, performance, and battery life of a phone. It also evaluates whether the model can effectively perform the arm-elimination phase with a single communication round when communication is expensive.

*4.4.2 Single-agent Distribution.* In this subcategory, the server serves as a global *agent*, and clients are regarded as the *environment*. Owing to the communication payload proportional to the size of items [5, 13], Khan et al. [72] propose an MAB-based optimization method named FCF-BTS

to alleviate the payload pressure. FCF-BTS computes rewards based on submitted item gradients and leverages Bayesian Thompson Sampling to formulate a sampling strategy. The central bandit model provides clients with an appropriate set of candidate items and hardly intervenes in local recommendation decisions. PrefFedPOI [217] adopts distillation-based RL to enhance the clustering capability in PFL-based HFedRec. The server maintains two clustering networks: one as a teacher giving guidance and the other as a policy-required optimization. The rewards are measured by local model performance, and the policy network improves itself by maximizing rewards obtained from both the environment and the teacher network.

*4.4.3 Summary and Challenges.* The Exploration-Exploitation dilemma of RL mirrors information cocoons and echo chambers [159] in personalized recommendation. People frequently receive content in line with their interests and opinions, but invisible to more diverse information. MAB-based RL gives HFedRec a fair strategy to tackle the polarization phenomenon and balance exploration and exploitation. However, some items may be popular in one environment but cannot illustrate their popularity in the whole federated context. Globally learned policy (e.g., UCB) tends to be suboptimal in heterogeneous environments [67]. The challenge behind this is *how agents adaptively fit the global policy according to local environment*, similar to PFL's proposition. In addition, recent practice focuses on the MAB problem, and thus another challenge is *how to embrace the broader ecology of RL*, such as deep RL [160] and reinforcement distillation [142] worth exploring in HFedRec.

## 5 HFEDREC FROM THE PRIVACY PERSPECTIVE

RecSys relies on users' preference to make accurate recommendations, but raw preference data may reveal sensitive information, even though metadata (e.g., names and phones) has been removed. Narayanan et al. [119] identify specific subscribers' records in the Netflix Prize dataset with background knowledge of the Internet Movie Database. Weinsberg et al. [172] leverage users' ratings in the Flixster and MovieLens datasets to infer their gender. In POI recommendation, check-in data may disclose user identity, health status, income level, and so forth [141, 184]. All of these studies account for the moment of privacy-preserving RecSys. Considering different privacy-preserving characteristics, we organize existing works of HFedRec in Table 4. The columns "Tech" and "Tar" account for the specific application target of a privacy protection technique.

### 5.1 Homomorphic Encryption-based Privacy Preservation

HE encrypts plaintexts with keys and supports calculations on ciphertexts without decryption [2]. Generally, HE can be distinguished from three dimensions:

(1) *Symmetry and asymmetry*: Symmetric encryption uses a common key for both encryption and decryption, whereas asymmetric encryption requires a public-private key pair.

(2) *Addition, multiplication, and hybrid*: Additive and multiplicative homomorphism maps ciphertext calculations to add. and mult. on plaintexts, respectively. Hybrid homomorphism has both abilities.

(3) *Partially, Somewhat, and Fully HE*: Partially HE supports an unlimited number of a single operation (i.e., add. or mult.); Somewhat HE supports multi-type operations of limited number; and Fully HE owns both benefits and supports an unlimited number of multi-type operations.

In terms of the third dimension, we categorize related works into partial and full homomorphism in the next subsections.

*5.1.1 Partial Homomorphism.* From our observation, current studies on HFedRec mainly exploit Paillier HE [124] to protect transmitted gradients or model weights. Paillier HE is an

Table 4. Representative Works in HFedRec from the Privacy Perspective

| Publication | Type | Tech | Tar | Loss | Publication | Type | Tech | Tar | Loss |
|---|---|---|---|---|---|---|---|---|---|
| FCMF [185] | M | PHE$^*$, AN$^\dagger$ | G$^{\dagger *}$, MW$^*$ | True$^\dagger$ | FMSS [94] | M, D | SS$^*$, FM$^\dagger$ | G$^*$, RD$^\dagger$ | True$^\dagger$ |
| FedMF [13] | M | PHE | G, MW | False | FedNCF [129] | D | SS | MW | False |
| FedNewsRec [133] | D | AN | G | True | Federated CF [162] | M | SS$^*$, GC$^\dagger$ | MW$^*$, RD$^\dagger$ | False |
| DP-PrivRec [163] | T | AN | G | True | SharedMF [196] | M | SS | G | False |
| FeSoG [102] | D | AN$^*$, FM$^\dagger$ | G$^*$, RD$^\dagger$ | True | FedRec$_{\text{L}}$ [92] | M | FM | RD | True |
| FedPerGNN [174] | D | PHE$^*$, AN$^\dagger$, FM$^\ddagger$ | RD$^{*\ddagger}$, G$^\dagger$ | True$^{\dagger\ddagger}$ | Federated MF [32] | M | PHE$^*$, RR$^\dagger$ | RD$^\dagger$, G$^*$, MW$^*$ | True$^\dagger$ |
| FedNCF [65] | D | AN | G | True | FedIS [30] | T | RR | RD, MW | True |
| FedRec++ [91] | M | FM | RD | False | FedPOIRec [128] | D | FHE, SS | MW | False |
| SDCF [64] | M | RR | RD | True | FedSeqRec [80] | D | AN | MW | True |
| Efficient-FedRec [193] | D | SS | G | False | KG-FedTrans4Rec [171] | D | AN | MW | True |
| FedGME [127] | D | PHE | MW | False | DFSR [103] | M | AN | G | True |
| FedHGNN [183] | D | RR$^*$, AN$^\dagger$, FM$^\ddagger$ | RD$^{*\ddagger}$, G$^\dagger$ | True$^{\dagger\ddagger}$ | SemiDFEGL [135] | D | AN$^*$, FM$^\dagger$ | MW$^*$, RD$^\dagger$ | True |
| FMFSS [226] | M | SS | G | False | FINDING [197] | D | PHE, FHE | G, MW | False |

**Type:** M: Statistical Machine Learning, D: Deep Learning, T: Meta Learning, R: Reinforcement Learning.
**Tech (privacy-preserving techniques):** PHE: Partially Homomorphic Encryption [124], FHE: Fully Homomorphic Encryption [25], AN: Additive Noise [35], RR: Randomized Response [38], FM: Fake Marks [94], SS: Secret Sharing [144], GC: Garbled Circuit [190].
**Tar (the applied targets of Tech):** RD: raw Rating Data, G: Gradients, MW: Model Weights such as item embeddings and transformation matrices.
**Loss (performance degradation):** If the introduction of privacy-preserving techniques degrades model performance, "Loss" is set as "True."

Note: $*$, $\dagger$, and $\ddagger$ indicate which target a privacy-preserving technique is applied to.

*asymmetric*, *additive*, and *partial* encryption schema. Based on Paillier HE, FedMF [13] first defines the encrypted training steps for secure MF:

— *Step 1*: KeyGen $\rightarrow (pk, sk)$—An honest client generates a public-private key pair of $(pk, sk)$, where $pk$ is distributed to all parties and $sk$ is only held by clients.
— *Step 2*: Enc$(\boldsymbol{V}, pk) \rightarrow C_{\boldsymbol{V}}$—The central server encrypts the initialized item embedding matrix $\boldsymbol{V}$ with $pk$ to obtain $C_{\boldsymbol{V}}$, and it is available for download by all clients.
— *Step 3*: Dec$(C_{\boldsymbol{V}}, sk) \rightarrow \boldsymbol{V}$—Each client downloads the latest encrypted matrix $C_{\boldsymbol{V}}$ from the central server and deciphers it with the $sk$ to obtain $\boldsymbol{V}$.
— *Step 4*: Clients locally update the user embedding and calculate items' gradients $\nabla_g$.
— *Step 5*: Enc$(\nabla_g, pk) \rightarrow C_{\nabla_g}$—Items' gradients are encrypted with $pk$ and sent to the server.
— *Step 6*: The central server can aggregate submitted gradients $C_{\nabla_g}$ and updates global item embeddings $C_{\boldsymbol{V}}$ via homomorphic properties:

$$\begin{aligned} \mathrm{D}(\mathrm{E}(m_i) \times \mathrm{E}(m_j)) &= m_i + m_j \\ \mathrm{D}(\mathrm{E}(m)^k (\mathrm{mod}\ n^2)) &= km (\mathrm{mod}\ n), \end{aligned} \tag{7}$$

where $\mathrm{E}(\cdot)$ and $\mathrm{D}(\cdot)$ are encryption and decryption algorithms, and $m_i$ and $m_j$ denote plaintexts.
— *Step 7*: Repeat steps 3 through 6 until convergence.

The following variants of federated MF [32, 185] adopt a similar procedure to protect data transmission. The preceding steps can be easily extended to deep learning based HFedRec [127] when the centrally aggregated data are embeddings or hidden layers' weights. Differently, FedPerGNN [174] designs an LGE schema to incorporate high-order connectivity, and keys are generated in the server. The schema introduces a third party to collect user embeddings and encrypted item IDs. Through matching item sequences, the third party can provide each user with user embeddings of her neighbors who have similar behaviors.

*5.1.2 Full Homomorphism.* Paillier HE only allows calculations of positive integers, and thus extra transformations are required to process float and negative numbers. To facilitate computational flexibility, FedPOIRec [128] utilizes Cheon-Kim-Kim-Song (CKKS) HE [25] to search nearest user embeddings as social neighbors. The social integration protocol is a deep promotion of FedPerGNN's and formulated as four detailed phases. As an *asymmetric*, *hybrid*, and *full* encryption schema, CKKS HE provides the protocol with additive and multiplicative operations on real numbers, making it practical to compute cosine similarities between user embeddings. Except securely averaging up loaded parameters by Paillier HE, FINDING [197] also employs Fully HE to protect $k$-means clustering on user representations from privacy disclosure [179]. Cluster-level models are then developed for PFL without privacy threats.

*5.1.3 Summary and Challenges.* Even though the encryption-decryption process may accumulate noise, the error fluctuation range is acceptable as evaluated in other works [13, 185]. Hence, HE guarantees both accuracy requirement and secure parameter exchange for HFedRec. Nevertheless, the key pair generation and frequent encryption-decryption process may lead to intolerable storage requirements, computational complexity, and communication overhead [13, 127, 128]. Owing to limited power of local devices, the intrinsic challenge is *how to trade off security and efficiency*, especially in the cross-user federation. When the server solely performs the parameter aggregation, some homomorphic properties are redundant and can be refined for efficiency enhancement [66].

## 5.2 Differential Privacy-based Privacy Preservation

DP aims to add extra noise and prevent computational outputs from being sensitive to the presence or absence of a specific data point [36], wherein central DP (CDP) requires a trusted third party to store data centrally, and statistical information without individual characteristics is responded to external query requests. Given privacy loss parameters $(\epsilon, \delta)$ and a randomized algorithm M, for any two neighbor datasets $X$, $X'$ (differing in a single entry $x \in X$, $X' = X \backslash \{x\}$) and output $O \in \text{Range(M)}$, $(\epsilon, \delta)$-DP is satisfied when

$$\Pr[\mathrm{M}(X) \in O] \leq e^{\epsilon} \Pr[\mathrm{M}(X') \in O] + \delta, \tag{8}$$

where a greater privacy budget $\epsilon$ means a greater difference between probability distributions of $\mathrm{M}(X)$ and $\mathrm{M}(X')$. The definition guarantees that $\mathrm{M}(X)$ and $\mathrm{M}(X')$ have similar probability distribution but permit a small deviation range $\delta$.

Since FL restricts raw data to be retained locally, we are not aware of the differences between users' datasets in the granularity of a single sample. The following subsections will depict specific DP techniques in the federated setting.

*5.2.1 Additive Noise.* The noises are real values from a certain independent distribution and added to a query function $f$. The basic form is

$$\mathrm{M}(\nabla_g) := f(\nabla_g) + \text{Noise}(\alpha, \beta), \tag{9}$$

where $\nabla_g$ are gradients submitted to the server for aggregation and Noise$(\cdot)$ denotes Laplace or Gaussian distribution. McMahan et al. [112] first propose user-level DP to ensure that whether a specific user is present or absent in the federation cannot be inferred by the server and external adversaries:

— *Step 1*: The server initializes model parameters and samples a batch of clients to start the federated training process.
— *Step 2*: Selected clients request and download model parameters from the server. Then they train their local model based on the local dataset for several epochs.

— *Step 3*: After clients complete local training in parallel, gradients are clipped and transmitted to the server.
— *Step 4*: The server aggregates clipped gradients according to bounded-sensitivity estimators. Then global model parameters are updated with aggregated gradients and Gaussian noise.
— *Step 5*: Repeat steps 2 through 4 until convergence.

In practice, applying norm clip can avoid potential gradient explosion [209] and bound each individual sample's influence on aggregated results [1]. Based on the preceding process, DP-PrivRec [163] draws noise from the $(0, \sigma^2)$ Gaussian distribution to implement noisy stochastic gradient descent. A numerically stable moment accountant is employed to calculate the privacy budget $\epsilon$. Different from adding noise to the server-side aggregation, most HFedRec [80, 133, 135, 171, 174, 183, 185] choose to add local noise to gradients or model weights before submission, which satisfies the privacy guarantee of **Local DP (LDP)**. Despite involving perturbation in each client and increasing the total amount of noise, LDP accomplishes fine-grained privatization of local data. Considering gradients varying in magnitudes, FeSoG [102] and DFSR [103] devise the adaptive noisy strength based on $mean(\nabla_g)$, and FedNCF [65] further sets a different threshold to clip gradients.

*5.2.2 Randomized Response.* RR protects original data through the uncertainty when responding to sensitive topics [169]. In other words, for ID-sensitive rating behaviors, when user $i$ interacts with item $j$, there is a probability to reverse the fact. Some studies [32, 64] perturb the user-item interaction matrix by two-stage randomization including a permanent RR (PRR) and an instantaneous RR (IRR). PRR is to perturb the binary matrix before model training and thus signed as "permanent," whereas IRR is applied to the binary matrix at each communication round. Similarly, FedIS [30] first obfuscates real interaction labels and then performs sparsification of model updates with the Bernoulli distribution. Due to the perturbation enlargement phenomenon [212] in HINs, FedHGNN [183] utilizes exponential mechanism and degree-preserving RR [51] to prevent a denser perturbed graph. The server can reconstruct a global HIN from local perturbed results while ensuring privacy security.

*5.2.3 Summary and Challenges.* Compared with HE, DP has lower computational complexity and can prevent privacy leakage while keeping higher efficiency, such as non-zero transmission in FedIS's sparsity perturbation [30]. However, noises are inevitably introduced into the training process and the model accuracy is inversely proportional to the degree of privacy preservation [80, 171, 174]. The intrinsic challenge is *how to trade off security and accuracy*, and it is expected to attach rigorous theorization for the guarantee of $(\epsilon, \delta)$-DP in both training and online serving stages [97]. Furthermore, due to the intuitive composition theorems and compatibility with Gaussian noise, **Rényi DP (RDP)** [116] is increasingly receiving much research attention [22, 56, 170]. DGREC [227] pioneers the investigation of RDP-based secure gradient sharing with tighter bounds on the privacy budget. More refined and quantitatively tractable analyses on privacy loss remain to be studied.

### 5.3 Secure Multiparty Computation-based Privacy Preservation

SMC is a general cryptographic scheme where a group of participants cooperatively compute an objective function without exposing their private data in a distributed environment [220]. Participants solely acquire their own corresponding computational result (if needed) and have no consciousness of other information. SMC first entered the public view through the work of Yao [190] and has incorporated multiple cryptographic tools after decades of development, such as the SS protocol and GC protocol detailed next.

*5.3.1 Secret Sharing.* SS means that a dealer fragments private data called *secret* into pieces of *shares* distributed among participants [144]. The original secret can be reconstructed by combining sufficient shares. In the federated setting, the split secret generally refers to publicly transmitted model parameters. To prevent secrets from divulging privacy to an honest-but-curious server or attackers, the *Secure Aggregation* problem [150, 176] arises for a more trustworthy aggregation schema. The server is blind to what clients submit and only acquires aggregated results.

An intuitive practice is secret splitting [3], where each client divides a gradient $\nabla_g$ (or model weights) into $n$ parts and all subgradients are required to recover $\nabla_g = \sum_{i=1}^{n} \nabla_{g_i}$. Local item gradients are first divided into several random numbers, of which one share is retained locally and then the rest are distributed to others [196, 211]. If there are clients $A$ and $B$ to upload their gradients $X$ and $Y$, respectively, the sharing steps are as follows:

— *Step 1*: Clients $A$ and $B$ randomly divide their gradient into $\{x_1, x_2\}$ and $\{y_1, y_2\}$, respectively.
— *Step 2*: Clients $A$ and $B$ distribute their subgradients to each other. Client $A$ holds $\{x_1, y_1\}$, and client $B$ holds $\{x_2, y_2\}$.
— *Step 3*: Both sum locally stored subgradients—that is, $z_1 = x_1 + y_1$ in client $A$, and $z_2 = x_2 + y_2$ in client $B$. $z_1$ and $z_2$ are uploaded to the central server.
— *Step 4*: The server gets the aggregated gradient $Z = z_1 + z_2$.

Lin et al. [94] further propose a generic cross-user HFedRec framework that examines more MF variants and neural networks under the preceding SS process. As a single update gradient is divided into multiple shares, the server may misestimate the real gradient number, leading to suboptimal averaging results. FMFSS [226] simultaneously leverages the splitting operation to securely derive user-agnostic item interaction frequencies at the server side.

However, forcing all shares to reconstruct the original information is impractical when clients join in and drop out of the federation dynamically (i.e., some shares may be lost before aggregation). More robust SS protocols are necessary to deal with the unpredictable client activities. One is the threshold SS scheme of Shamir [144]. The threshold controls the number of shares entailed in recovering the secret. Bonawitz et al. [11] propose a well-designed secure aggregation schema based on Shamir's SS. It meets multiple requirements, including secure summation, constant communication rounds, low communication overhead, and failure robustness. Many HFedRec [128, 129, 188, 193] are motivated by their work to implement parameter aggregation in a secure manner. To hide user-item interaction behaviors, Efficient-FedRec [193] also utilizes Bonawitz's solution to unite news sets from a user group before the gradient aggregation.

*5.3.2 Garbled Circuit.* GC initially involves secure two-party computation in the protocol of Yao [190]. The word *circuit* means that parties jointly compute a public function expressed as a logic circuit without exposing individual input data. The word *garbled* refers to the garbled truth table in which each encrypted entry is randomly ordered and embodies inputs and the corresponding output of a gate [10]. Subsequent GC protocols are dedicated to decreasing communication and computation costs [76, 202] and generalizing the scale to multi-parties [9, 186]. From our observation, only FederatedCF [162] explores GC and involves two organizations, of which one undertakes the garbler and the other undertakes the evaluator. It forms the classic two-party GC schema and performs Yao sharing on user behaviors.

*5.3.3 Summary and Challenges.* SMC aims at multi-party privacy security and is naturally adaptive to HFedRec. The secure aggregation schema can generate equivalent results as vanilla aggregation [129]. One growing concern is *how to reduce the multi-party communication cost* (e.g., data volume, time, rounds) as the client size increases [128, 129, 193]. Moreover, recent HFedRec

only explore the tip of the iceberg in the SMC field. Other protocols, such as SS for confronting dishonest dealers [40] and GC for defending malicious adversaries [95], can also remain as challenges.

### 5.4 Fake Marks

Fake marks is a light-obfuscation approach that samples unrated items as fake user-item interactions to prevent the disclosure of personal behaviors. Concretely, given user $i$, unrated items $\boldsymbol{v}'_j$ are locally sampled from $V'_{u_i} = V \backslash V_{u_i}$ and set virtual ratings $r'_{ij}$ for training. The server is only aware of whether an item $v_j$ is in $\{V_{u_i} \cup V'_{u_i}\}$ rather than real interactions $V_{u_i}$. FedRec$_{\text{L}}$ [92] assigns virtual ratings through user averaging (UA) and hybrid filling (HF) strategies:

$$r'_{ij} = \begin{cases} \overline{r}_{u_i} = \dfrac{1}{|V_{u_i}|} \displaystyle\sum_{v_k \in V_{u_i}} r_{ik} & \text{, UA} \\ \hat{r}_{ij} = \boldsymbol{u}_i \boldsymbol{v}'_j & \text{, HF} \end{cases} \tag{10}$$

where $|V_{u_i}|$ is the number of items rated by user $i$. The UA strategy computes the average ratings of rated items, and the HF strategy uses the inner product of $\boldsymbol{u}_i$ and $\boldsymbol{v}'_j$. FeSoG [102] also gives predicted ratings to fake items and empirically proves its robustness. Based on FedRec$_{\text{L}}$, FedRec++ [91] designs a denoising approach to alleviate the performance degradation brought by fake items. Selected denoising clients gather fake items' gradients from ordinary clients. When the server aggregates gradients from two types of clients, the noise can be eliminated. Instead of assigning fake ratings, FedPerGNN [174] and SemiDFEGL [135] directly generate fake items' gradients, depending on the Gaussian distribution of real item gradients. FMSS [94] further combines fake marks with secret splitting to protect gradient transmission.

### 5.5 Malicious Attack and Byzantine Resilience

Despite the federated security augmented by privacy-preserving techniques, user privacy and recommendation performance are still susceptible to malicious attacks [24]. Parties in the preceding HFedRec are passive to obey the promissory federated process; however, even if a small percentage of active adversaries tamper with the federation, stealthy and harmful attacks can compromise the model validity and private user data. We sort out the attack and defense studies in Table 5. Except for "Special Attack Assumes," there are *common assumes* for all attackers:

(1) They can compromise a small proportion of clients if attacks are from malicious clients.

(2) They have prior knowledge of a candidate item pool, when the item embedding matrix is transmitted between the server and clients.

(3) They have no access to benign clients' local data, such as interacted items and user embeddings. Whether malicious clients can manipulate their own private data depends on specific attacks.

In the following subsections, we trace recent progress from poisoning and inference attacks.

*5.5.1 Poisoning Attack.* We conclude two aims for poisoning attack. The first aim is "ItemPromotion" to boost target items' exposure rate without side effects on the model performance. This action is named *Targeted Poison*, which shifts users' interests toward items specified by adversaries. Due to the popularity bias in recommended items, PipAttack [216] conceals target items through the popularity obfuscation. Concretely, an additional classification model is trained to label each item with a popularity level. Its goal is to align target items with top-popular items. FedRecAttack [139] utilizes publicly available user-item interactions to approximate user embeddings. The experiment exhibits that 3% of malicious clients and 1% of public interactions can remain high

Table 5. Representative Works of Attack and Defense

| Publication | Type | Model | Constraints | Special Attack Assumes |
|---|---|---|---|---|
| FedRecAttack [139]<br>**Aim:** ItemPromotion | Targeted Model Poison on Uploaded Gradients | BPRMF with LDP | limit the maximum l2-norm and number of poisoned item gradients | **access** the model structure (e.g., hidden layers' weights and item embeddings) except all user embeddings<br>**access** some hyperparameters (e.g., learning rate)<br>**have** a small fraction of public user-item interactions<br>**alter** malicious clients' gradients |
| PipAttack [216]<br>**Aim:** ItemPromotion | Targeted Model Poison on Uploaded Gradients | NCF | limit the distance between poisoned gradients and the original ones | **access** the model structure except benign user embeddings<br>**have** clues on the popularity of items<br>**alter** malicious clients' local models and their gradients |
| A-ra & A-hum [138]<br>**Aim:** ItemPromotion | Targeted Model Poison on Uploaded Gradients | NCF | – | **access** the model structure except all user embeddings<br>**alter** malicious clients' gradients |
| PSMU [199]<br>**Defense:** HiCS<br>**Aim:** ItemPromotion | Targeted Model Poison on Uploaded Gradients | NCF and LightGCN | only upload the target items' poisoned gradients | **access** the model structure except benign user embeddings (malicious clients are synthetic; read-write privileges, like local data and user embeddings, are naturally owned) |
| FedAttack [175]<br>**Aim:** ValidityFailure | Untargeted Data Poison on Pos./Neg. Samples | NCF and BERT4Rec | – | **access** local user and item embeddings<br>**have** normal user profiles to infer user embeddings<br>**alter** malicious clients' local model inputs and labels |
| ClusterAttack [198]<br>**Defense:** UNION<br>**Aim:** ValidityFailure | Untargeted Model Poison on Uploaded Gradients | BPRMF and SASRec | clip poisoned gradients with a norm bound estimated from normal gradients | **access** the model structure except all user embeddings<br>**have** normal interactions to clip poisoned gradients<br>**alter** malicious clients' gradients |
| IMIA [201]<br>**Defense:** RegTerm<br>**Aim:** PrivateData | Membership Inference on User-Item Interactions | NCF and LightGCN with LDP | – | **access** the target clients' uploaded parameters in accord with pre-agreed federated protocol<br>**access** some hyperparameters (e.g., negative sampling rate and learning rate) |
| EIFedMF [14]<br>**Aim:** PrivateData | Defense against Attribute Inference Attack | PMF with LDP, HE, and OT | – | – |

**Aim:** ItemPromotion: the attack aims to popularize target items as many users as possible, ValidityFailure: the attack aims to degrade the model performance, PrivateData: the attack aims to infer private user data (interacted items in IMIA and user profiles in EIFedMF).

**Defense:** The proposed defense method to address the corresponding attack.

**Type:** Attack and defense types.

**Model (victim example):** BPRMF: Bayesian Personalized Ranking Based MF [137], NCF: Neural Collaborative Filtering [50], LightGCN: Light GCN [49], SASRec: Self-Attentive Sequential Recommendation [71], BERT4Rec: BERT-Based Recommendation [152], PMF: Probability MF [143], OT: Oblivious Transfer [118].

**Constraints:** To prevent the attack from being detected (i.e., stealthy) and affecting the model performance.

**Assumes**: Prior knowledge of each attack model.

effectivity. Subsequent studies perform attacks with less prior knowledge. Rong et al. [138] randomly approximate user embeddings from a Gaussian distribution, but the attack performance lacks stability. Therefore, PSMU [199] constructs synthetic clients with fake interactions, and target items highly ranked in synthetic clients also have a popularity bias on benign clients. It amplifies the influence of target items in similar substitutes and top-$K$ recommendations. The corresponding defense method performs hierarchical gradient clipping and sparsified updating in the server.

Another aim is "ValidityFailure" to undermine the federated model performance. This action is named *Untargeted Poison*, which affects users' normal experience of recommendation services. ClusterAttack [198] first applies $k$-means to item embeddings received from the server. A within-cluster variance loss is then computed to minimize the discrepancy between item embeddings and

their centroids, consequently perturbing the actual item embedding distribution and degrading the model validity. Hence, its defense method adopts a contrastive task to sustain the uniform distribution of item embeddings. All of the preceding poisoning attacks are *Model Poison* to alter gradients uploaded to the server. Contrarily, FedAttack [175] belongs to *Data Poison*, which reverses labels of positive and negative items to subvert HFedRec.

*5.5.2 Inference Attack.* Inference attack aims to extrapolate private information in training samples [55]. From our observation, recent inference attacker refers to the semi-honest server curious about private user data, while poisoning attackers are malicious clients. IMIA [201] first constructs a shadow dataset that labels of uploaded item embeddings are proportionally assigned by the negative sampling rate. Then, if an item embedding from the trained shadow model is similar to the uploaded one in Euclidean space, the *Membership Inference* on user behaviors is hit. The local defender utilizes a regularization term to limit the update of received public parameters. For defensing user profiles (e.g., age, gender), EIFedMF [14] combines multiple privacy-preserving techniques against the *Attribute Inference*. Full homomorphism of CKKS [25] is exploited to cluster user-item interactions, and then LDP bounds the probability whether one user is in a particular cluster. Finally, OT [118] obfuscates one user's item ratings from his own cluster to another.

*5.5.3 Summary and Challenges.* Recent attack methods have achieved remarkable effectivity on subverting the reliability and confidentiality of HFedRec. For instance, PSMU [199] can reach 1.0 Exposure Rate@5 with only two or three malicious clients on MovieLens-1M. IMIA [201] empirically reveals that LDP has a privacy-preserving impact when the additive noise largely compromises recommendation performance. Therefore, *there is an imbalance between existing privacy protection efforts and attack scenarios in HFedRec*. Apart from privacy discussions, simulation experiments against malicious attacks are expected to prove the validity of privacy techniques.

## 6 HFEDREC FROM THE FEDERATED PERSPECTIVE

By this point, we believe that you already have a comprehensive grasp of model design and privacy protection in this field. Now, we emphasize two critical issues from the federated perspective: communication optimization and fairness perception.

### 6.1 Communication Optimization for HFedRec

The ever-growing scale of users and items accounts for the sparsity issue of RecSys [194]—that is, some user-item interactions are insufficient to cover the broad user groups and candidate items. It is imperative to involve more clients in the federation to sustain the recommendation performance; however, resource-constrained clients (e.g., low bandwidth) and the single-point server may become a bottleneck to accelerate the federated model convergence. In this section, we concentrate on communication-efficient techniques, as shown in Table 6. Specific model names are not listed as in Table 5, because some techniques have no inherent correlation with neural network structure.

*6.1.1 Knowledge Distillation.* The goal of "distillation" is to allow a smaller *student* model to exploit supervised signals from another *teacher* model to acquire similar or superior performance [45]. The "knowledge" can be hidden layers' weights (termed *feature-based knowledge*) or soft logits inferred by the last prediction layer (termed *response-based knowledge*). For the latter knowledge, given a user $u_i$ and interacted items $V_{u_i}$, the distillation loss is typically formulated as

$$l_{distill} = \mathrm{KL}\left(\mathrm{Logit}_t(u_i, V_{u_i}|\theta_t), \mathrm{Logit}_s(u_i, V_{u_i}|\theta_s)\right), \tag{11}$$

where $\mathrm{KL}(\cdot)$ denotes the Kullback–Leibler divergence, and $\mathrm{Logit}_t(\cdot)$ and $\mathrm{Logit}_s(\cdot)$ predict logits of candidate items through the teacher and student models, respectively. FedKD [173] locally holds

Table 6. Representative Works of Communication-Efficient HFedRec

| Publication | Type | U/O | Tech | Tar | Publication | Type | U/O | Tech | Tar |
|---|---|---|---|---|---|---|---|---|---|
| FedKD [173] | D | O | KD†, Comp* | Model†, Param* | URFL [225] | D | U | Edge | Distr |
| ECCT [88] | D | O | KD | Model | LightFR [208] | M | U | Comp | Model, Param |
| JointRec [33] | D | U | Edge†, Comp* | Distr†, Param* | DGREC [227] | D | U | Comp | Param |
| HFCCD [23] | M | U | Edge†, Comp* | Distr†, Param* | CF-FedSR [104] | D | U | Cluster | ItemSequence |
| HFSA [86] | D | U | Edge | Distr | FedFast [117] | D | U | Cluster | UserMeta |
| CCF-CRR [96] | D | U | Edge | Distr | PerFedRec [105] | D | U | Cluster | UserEmbed |
| GIMA [59] | D | U | Edge | Distr | SemiDFEGL [135] | D | U | Cluster | Graph, ItemEmbed |

**Type:** M: Statistical Machine Learning, D: Deep Learning.**U/O:** U: Cross-User Federation, O: Cross-Organization Federation.
**Tech(communication-efficient techniques):** KD: Knowledge Distillation [53], Comp: Parameter Compression, Edge: Edge Computing [58], Cluster: Cluster-based Client Sampling.
**Tar(the applied targets of Tech):** Model: Federated Model, Param: Transferred Parameters, Distr: Distribution Architecture, UserEmbed: User Embedding, ItemEmbed: Item Embedding, ItemSequence: Encoded Item Sequences, Graph: Encoded Local Graphs, UserMeta: User Metadata.

Note: ∗ and † indicate which target a communication-efficient technique is applied to.

a large teacher model which also learns response- and feature-based knowledge from the global student in a mutual-distillation manner. Only the student model with small parameter size is globally aggregated and shared, thus decreasing the communication cost. Furthermore, the strength of distillation losses is adaptively computed by loss descent degrees. Oppositely, ECCT [88] puts the large teacher model in the server assuming that non-private and centralized features are available. Clients only sustain small student models, and prediction logits with low communication burdens are transmitted for optimizing both sides' recommendation performance.

*6.1.2 Parameter Compression.* The presence of massive user-item interactions can result in an increasing model size, inevitably posing a challenge for resource-constrained clients in handling heavy communication and intensive computing [168]. Hence, parameter compression is intent on (i) decreasing uplink and downlink burden during the client-server transmission and (ii) enhancing computational efficiency in both training and inference phases.

LightFR [208] is a noteworthy work satisfying the two intents. It binarizes continuous user/item embeddings into the discrete Hamming space by learning to hash [161]. Local discrete optimization and global discrete aggregation modules are tailored to update binary user vectors and the item matrix. The discrete Hamming space saves computing complexity and eight times communication consumption compared with Euclidean space. FedKD [173] factorizes transmitted gradients of the student model into three smaller matrices through singular value decomposition [44]. JointRec [33] combines low-rank MF and 8-bit quantization to compress uplink model weights. HFCCD [23] utilizes Count Sketch [15] to map high-dimensional gradients to a fixed-size sketch. Referring to the design of a one-bit encoder [29], DGREC [227] encodes each element of transmitted gradients as −1 or 1, which minimizes the communication cost and simultaneously satisfies RDP. The applied compression schemes in other works [23, 33, 173, 227] also require decompression/decoding before parameter aggregation or other operations.

*6.1.3 Edge Computing.* The Internet of Everything (IoE) era is continually driving the integration of terminal devices to unlock unprecedented value beyond their individual capabilities, which is equally observable in the federated distribution environment [77]. It is intractable for a single server to ensure the immediacy of recommendation services when connected devices gradually approach its bandwidth and computing limits. With edge servers deployed closer to end users, edge computing can expand the cloud infrastructure and realize low-latency communication [33, 58].

One dominant hierarchical architecture is cloud-edge-end collaboration [219]. As a bridge between the central server and clients, edge clusters aim to divert long-distance communication consumption and partial data processing. HFSA [86] proposes a semi-asynchronous aggregation mechanism and decides the cycle of edge-end collaboration by local training efficiency—for example, clients too slow to catch up on others' progress are only permitted to receive data from edge servers. CCF-CRR [96] probes into the cruising route recommendation for taxis and passengers. An edge server is responsible for training a federated recommendation model with taxis in the same region. Receiving predicted passenger waiting time from edge servers, the cloud server ensures the taxi demand-supply equilibrium among regions. HFCCD [23] introduces a compression scheme [15] and DP to improve the reliability and confidentiality of multi-party collaboration. **Mobile Edge Computing (MEC)** [167] is another distributed architecture where computing, storage, and network resources are strategically assigned closer to mobile terminals or edge devices. For inspiring users to participate in MEC-based HFedRec, GIMA [59] proposes an incentive mechanism by the two-stage Stackelberg game [182]. The profit-oriented competition among users consequently accelerates model convergence. Due to the spatial-temporal varying subjectivity (e.g., interests) of mobile users, URFL [225] devises an unsupervised recurrent FL to predict global content popularity. It sacrifices a tolerable rise in data traffic within the MEC network while improving recommendation accuracy.

*6.1.4 Client Cluster.* Different user interests are non-IID among clients such that random client sampling before a communication round may neglect co-occurrence preference patterns of part user groups. The objective of client cluster is to proportionally sample representative clients (i.e., delegates) from each community, thus reducing intra-cluster non-IIDness and accelerating the federated training.

FedFast [117] clusters clients by summary statistics of user profiles, then selects delegates from each cluster to join the current round. Its aggregation strategy shares the trained weights of delegates with others. The convergence speed is remarkable such that non-cluster aggregation endures 94% more communication rounds to match FedFast's performance. Similarly, Cali3F [231] applies FedFast's practice to fairness-aware HFedRec. CF-FedSR [104] clusters concatenated vectors of short- and long-term preference encoded by GRU4Rec [52], whereas PerFedRec [105] and PerFedRec++ [106] utilize user embeddings to perform clustering. Apart from the server sampling representative clients, SemiDFEGL [135] also empowers a client to communicate with others in the same cluster. The P2P collaboration facilitates the cross-client feature diffusion and decreases the number of heavy cloud-end communications. Note that there are more studies adopting the cluster-based federation, and HFedRec benefits from other respects such as privacy preservation [14], overall recommendation performance [61, 192, 214, 217], and the cold-start issue [197].

*6.1.5 Summary and Challenges.* The preceding communication-efficient techniques exchange a diversity of viewpoints on model size, transmission, distributed architecture, and client sampling. Besides, *special learning paradigms* also have a significant reduction on communication cost. For instance, meta learning based FedIS [30] and FedMeta [17] consume fewer convergence epochs and transmission quantity than vanilla aggregation. Reinforcement-based FCF-BTS [72] optimizes the downlink payload by a bandit model in the server. EfficientFedRec [193] puts single user models locally as split learning. Their details were described in Section 4. From our point of view, the federated communication issue should be emphasized in the real IoE scenario, since HFedRec naturally applies to geographically dispersed end devices [62]. Therefore, one challenge is *how to incorporate load balancing, service discovery, and fast adaption in edge computing empowered HFedRec* [48, 153]. In the future, electric power, energy utilization, storage, and more hardware resources can be examined to bring HFedRec into everyone's life.

### 6.2 Fairness Perception for HFedRec

Ideally, preference data generated by every individual or group (varying in gender, occupation, age, etc.) should have an equal opportunity to acquire collaborative training from the federated model. The recommendation behaviors that omit disadvantaged groups or only highlight popular tastes are unfair actions with discrimination and prejudice [114]. Although it is confined to the horizontal federated recommendation scenario, there is still no universal fairness definition capable of encapsulating all fairness objectives [19]. We classify existing efforts into client and group levels.

*6.2.1 Client-level Fairness.* Current client-level fairness is devoted to pursuing uniform accuracy among clients. Given client $C$ and two federated models $f_1$ and $f_2$, $f_1$ is fairer than $f_2$ when

$$\text{std}\left(\{\text{Eval}(f_1)_c\}\right) < \text{std}\left(\{\text{Eval}(f_2)_c\}\right), c \in C, \tag{12}$$

where std$(\cdot)$ denotes the standard deviation of clients' metrics Eval$(\cdot)$ on test data. There are other uniformity functions that measure the performance distribution, such as cosine similarity and entropy [84]. We refer readers to the work of Chen et al. [19] for more fairness notions.

For server-side unfairness, the classical aggregation strategy weighs the importance of uploaded parameters by the size of local datasets. Clients with sufficient local data may bias the global recommendation model to their preference patterns. Hence, one fair solution is to *reweight* the parameter aggregation. FL-GMT [87] collects clients' training losses in each communication round. The larger the training loss, the greater the weight coefficient. Its core idea is that a larger local loss represents poorer local model performance, so more resources should be invested. Similarly, CF-FedSR [104] generates weight coefficients for the aggregation stage based on the combination of local performance metrics and data size.

For the client-side unfairness, local model parameters are typically substituted with the global aggregated results, which equivalently guides all clients to the generalized preference pattern. The global update direction may be quite distinct from local ones. To this issue, Cali3F [231] additionally introduces a *regularized term* into the loss function to control the differentiated degree between local and global parameters. The preceding fairness effects are usually verified by variance analysis in comparison with baselines.

*6.2.2 Group-level Fairness.* The group-level fairness aims to mitigate performance bias against protected disadvantaged or minority groups. The fairness definition varies depending on how groups are divided and what fairness notions are desired. F2MF [98] encourages that the discrepancy of group-average metrics should be minimized within each group pair. The fairness loss is jointly optimized with the target recommendation task as *multi-task learning*. For blinding participants to the attached group of each client, DP is utilized to obfuscate individual information. Analogously, given two client groups $C_1$ and $C_2$, FPFR [166] defines the fairness loss as

$$L_{fair} = \left| \frac{1}{|C_1|} \sum_{c \in C_1} \text{Eval}(f_c) - \frac{1}{|C_2|} \sum_{c' \in C_2} \text{Eval}(f_{c'}) \right|, \tag{13}$$

where $f_c$ denotes the local personalized model of client $c$ and FPFR selects NDCG@10 as Eval$(\cdot)$. The client groups are clustered by $k$-means on user embeddings. According to different system capabilities, RF$^2$ [109] groups clients into low, mid, and high tiers. It provides a platform to evaluate federated fairness across tiers when system-induced data heterogeneity exists.

*6.2.3 Summary and Challenges.* It is expected that the empowerment of fairness perception will not cause adverse effects on the overall performance of federated models. The empirical study

in F2MF [98] indicates a tradeoff between fairness and accuracy. Besides, the evolutionary users' interests account for the challenge that *groups divided by static attributes cannot completely represent disadvantaged or minority parties*. Simultaneously, the dynamic data heterogeneity induced by client sampling and non-IIDness also limits local debiasing efforts on the fair performance distribution [39]. More client- and server-side biases remain to be considered.

## 7 OPEN ISSUES AND FUTURE DIRECTIONS

It is worth pointing out that federated research has not penetrated into all aspects of RecSys. In this section, we list several open issues and promising directions.

### 7.1 A Benchmark Is What We Need

In the centralized recommendation, basic experimental settings and performance comparison have been systematically established in generally acknowledged benchmarks, such as RecBole [222, 223] and FuxiCTR [229, 230]. However, as shown in Appendix B, we can see different programming languages, machine learning libraries, and federated environments applied in existing open source projects of HFedRec. Some baseline recommendation models (e.g., MF [57, 137, 143], NCF [50], and LightGCN [49]) are widely adopted but lack a unified federated setting to regulate non-IID data, client sampling, client size, global aggregation strategy, and so on. Since a series of recommendation models can be straightforwardly nested with federated training steps as listed in Section 4, a new benchmark for HFedRec (or FedRec) is urgently needed to fairly compare model performance, communication costs, fairness, and other fundamental issues.

### 7.2 Exploration of Heterogeneous FL

Most existing HFedRec implicitly require that the local recommendation model architecture across all participants is homogeneous, which significantly limits the application scenarios in the real world. For example, resource-constrained clients have to be excluded from the federated environment, if the collaboratively trained model continues to be sophisticated and exceeds their computational capabilities. To this issue, heterogeneous FL has emerged as a feasible solution [191]. Different clients can hold different models varying in size and architecture, known as model heterogeneity. Recent works transmit various types of knowledge carriers (e.g., logits on public datasets [79], additional homogeneous models [134, 146], and class prototypes [156]) to achieve knowledge sharing across local heterogeneous models. Unfortunately, these works cannot be directly transferred to the horizontal federated recommendation due to different data structure, model architecture, and transmitted parameters. HeteFedRec [200] is the pioneering work of model-heterogeneous HFedRec, which allows each client to hold item embeddings with varying sizes. However, HeteFedRec still requires each client to keep the same local model architecture. Continuing to explore model-heterogeneous HFedRec and other heterogeneous forms is a promising research topic and essential for real-world applications.

### 7.3 Beyond Plain Graph with Knowledge

If side information of users or items is available, a KG is expected to enrich node representations and enhance recommendation performance [46]. The KG is a heterogeneous graph composed of triplets (*head*, *relation*, *tail*). Different KGs in the same domain can be complementary to each other; however, due to data privacy and business revenues, KGs may be prohibited from transferring triplets to a data center. With the progress of federated KGs (FedKG), collaborative representation learning of KGs without privacy disclosure is on the stage. Chen et al. [21] improve global entity embeddings in a PS architecture, whereas Peng et al. [126] implement federated KG embedding

by the PATE-GAN model [68] in a decentralized manner. The fusion of HFedRec and FedKG is significant for gaining a better recommendation effect [171].

### 7.4 Efficient Encryption Techniques

SMC consumes more data exchanges among participants when multiple privacy protocols cooperate with each other [11, 14], and DP is not so resilient when confronted with existing malicious attacks [139, 201]. It seems that HE becomes the priority choice. Nevertheless, according to Jiang et al. [66], Paillier [124] and CKKS [25] inflate the plaintext size by approximately 0.2k and 173k times, respectively. The memory overflow risk and intensive computing for resource-constrained clients is a serious concern. BatchCrypt [203] is a batch encryption to concurrently encipher plaintexts and shorten the length of ciphertexts. When HE mainly takes effect in secure aggregation at the server side, Jiang et al. [66] design a faster HE named FLASHE to meet the minimum security and functionality requirements. Both BatchCrypt and FLASHE are implemented in FATE.[1] The present HFedRec with homomorphic cryptosystems have noticed the balance between efficiency and security. The preceding progress will open a new vision for HFedRec.

### 7.5 The Cold-Start Issue and Federated Scalability

Federated scalability generally refers to the impact of massive participating clients on computing resources, communication efficiency, and multi-party privacy [60, 213]. In the recommendation scenario, another implication is the cold-start trouble caused by the expanding scale of users and items. The user cold-start issue faces new users who have not interacted with enough items to portray their preference, whereas the item cold-start (ICS) issue copes with how to present new items to users. IFedRec [206] assumes that the server holds rich side information of the whole item set. When the ICS issue occurred, clients can utilize encoded item attributes to initialize local item embeddings. Wahab et al. [160] also define the ICS problem and design a credibility-based trust mechanism to assess whether clients give reasonable predicted scores to cold items. Based on the observation that most users have few interactions in real-world news recommendation scenarios, FINDING [197] runs $k$-means on user representations to cluster clients into groups, thereby benefiting cold users from cluster-level models. More federated cold-start setups remain to be investigated in the scope of federated scalability.

### 7.6 The Non-IID Issue and Personalized Federation

Non-IID data is an inevitable and challenging issue among federated models [83, 85, 224]. Before the presence of FL, RecSys had noticed the performance degradation and divergence caused by non-IID preference data [12]. For instance, a user's behaviors may be guided by friends or groups, and adjacent items in a consecutive sequence may be related to each other. As FL confines raw data to be kept locally, extreme non-IIDness is more common than that in the centralized mode, such as attribute skew, label skew, and quantity skew [228]. One of the effective solutions is PFL, which has attracted much attention [155]. There are data-based (e.g., client selection [104]), model-based (e.g., meta learning [187]), and similarity-based approaches (e.g., multi-task learning [82]) to empower the personalized ability of local models. The exploration of tackling the non-IID issue in HFedRec is still in its infancy. For example, clients trade off personalization and generality through a regularization term between the local model $\theta_{local}$ and the corresponding cluster model $\theta_{cluster}$ [192] (or the global model $\theta_{global}$ [231]), whereas some combine $\theta_{local}$ with $\theta_{cluster}, \theta_{global}$ via pre-defined weights [104–106, 166, 217], such as $\theta^*_{local} = \gamma\theta_{local} + (1-\gamma)\theta_{cluster}$. Beyond the foregoing vanilla

[1] https://fate.fedai.org

model interpolation methods, FINDING [197] is concerned with fine-grained personalization that the interpolation weights vary with the continuous training progress, and different neural layers have different personalized needs. Incorporating more personalized techniques is a promising direction to further enhance recommendation performance.

## 8 CONCLUSION

RecSys is a crucial information filtering and retrieval technology to provide users with personalized recommendations. Meanwhile, the risk of privacy breach is hidden in traditional recommendation, which promotes the evolution of FedRec. In this survey, we comprehensively summarize the development status of HFedRec, a dominant research subject of FedRec. The proposed taxonomy involves interdisciplinary perspectives—that is, they not only elaborate on recommendation models through statistical machine learning, deep learning, meta learning, and RL but also take privacy-preserving techniques and federated issues into account. Unique challenges are discussed at the end of each section. Furthermore, we come up with several promising research directions. In future work, we will pay more attention to VFedRec and TFedRec, and compare them with HFedRec in detail.

## APPENDICES

## A RECOMMENDATION PROBLEM DEFINITIONS

For the general recommendation, we give the point-wise definition as an example [20]. For the sequential recommendation, we refer to the formalisms in other works [136, 164]. Table 7 indexes different recommendation problems for Table 2.

*Definition 3 (General Recommendation).* The model $f(\theta)$ learns historical user-item interactions $(u_i, v_j, r_{ij})$ to capture user interests and predicts top-ranked items in $V \backslash V_{u_i}$ for $u_i$. It aims to minimize the following deviation $\delta(\cdot)$ between the predicted rating $f(u_i, v_j|\theta)$ and the ground truth $r_{ij}$:

$$\theta^* = \arg\min_{\theta} \sum_{u_i \in U, v_j \in V_{u_i}} \delta\left(f(u_i, v_j|\theta), r_{ij}\right).$$

*Definition 4 (Sequential Recommendation).* The model $f(\theta)$ learns interactions in chronological order $[v_{u_i,1}, \ldots, v_{u_i,j}]$ to capture user interests. It aims to determine the next sequence (or item) $\hat{V}^*_{u_i}$ that maximizes the score for $u_i$:

$$\forall u_i \in U,\ \hat{V}^*_{u_i} = \arg\max f(u_i, [v_{u_i,1}, \ldots, v_{u_i,j}]|\theta). \tag{14}$$

Table 7. Index Code of Recommendation Problems for Table 2

| Category 1 | Category 2 | Index Code |
|---|---|---|
| General Recommendation | Vanilla User-Item Interactions | G1 |
| | User-Item Bipartite Graph | G2 |
| | Heterogeneous Graph (e.g., social links between users) | G3 |
| Sequential Recommendation | Vanilla Chronological Order | S1 |
| | Spatial-Temporal POI | S2 |
| | Knowledge Graph | S3 |

Table 8. Open Source Projects of HFedRec

| Publication | Year | Type | Lang | Framework | Tech | Link |
|---|---|---|---|---|---|---|
| $FedRec_L$ [92] | 2021 | M | Java | – | FM | https://csse.szu.edu.cn/staff/panwk/publications/FedRec |
| FCMF [185] | 2021 | M | Java, Python | – | PHE, AN | https://csse.szu.edu.cn/staff/panwk/publications/FCMF |
| FedMF [13] | 2021 | M | Python | – | PHE | https://github.com/Di-Chai/FedMF |
| F2MF [98] | 2022 | M | Python | PyTorch | AN | https://github.com/CharlieMat/FedFairRec |
| FedRecAttack [139] | 2022 | M | Python | PyTorch | AN | https://github.com/rdz98/FedRecAttack |
| FMSS [94] | 2022 | M, D | Java, Python | PyTorch | SS, FM | https://github.com/LachlanLin/FMSS |
| ClusterAttack [198] | 2023 | M, D | Python | PyTorch | – | https://github.com/yflyl613/fedrec |
| FINDING [197] | 2023 | D | Python, C++ | PyTorch | PHE, FHE | https://github.com/yusanshi/FINDING |
| FedNewsRec [133] | 2020 | D | Python | Keras | AN | https://github.com/taoqi98/FedNewsRec |
| Limited Negatives [123] | 2021 | D | Python | TensorFlow | – | https://git.io/federated_dual_encoder |
| Efficient-FedRec [193] | 2021 | D | Python | PyTorch | SS | https://github.com/yjw1029/Efficient-FedRec |
| FedVAE [131] | 2021 | D | Python | Rectorch | AN | https://tinyurl.com/155zn06i |
| HPFL [177] | 2021 | D | Python | PyTorch | – | https://github.com/bigdata-ustc/hierarchical-personalized-federated-learning |
| FedKD [173] | 2022 | D | Python | PyTorch | – | https://github.com/wuch15/FedKD |
| FedAttack [175] | 2022 | D | Python | Keras | – | https://github.com/wuch15/FedAttack |
| FedSeqRec [80] | 2022 | D | Python | TensorFlow | AN | https://github.com/MuziLee-x/FedSeqRec |
| PipAttack [216] | 2022 | D | Python | PyTorch | – | https://github.com/rdz98/PoisonFedDLRS |
| $RF^2$ [109] | 2022 | D | Python | DeepCTR | – | https://github.com/facebookresearch/RF2 |
| PFedRec [205] | 2023 | D | Python | PyTorch | – | https://github.com/Zhangcx19/IJCAI-23-PFedRec |
| CPF-POI [192] | 2023 | D | Python | PyTorch | – | https://github.com/Leavesy/CPF-POI |
| PrefFedPOI [217] | 2023 | D, R | Python | PyTorch | – | https://github.com/Leavesy/PrefFedPOI |
| HN3S [211] | 2024 | D | Python | PyTorch | SS | https://github.com/LukeZane118/HN3S |
| FedIS [30] | 2023 | T | Python | TensorFlow | RR | https://github.com/XuanangD/FedIS |
| Fed2-UCB [147] | 2021 | R | Python | – | – | https://github.com/ShenGroup/FMAB |

**Type:** M: Statistical Machine Learning, D: Deep Learning, T: Meta Learning, R: Reinforcement Learning.
**Tech (privacy-preserving techniques):** PHE: Paillier Homomorphic Encryption, AN: Additive Noise, FM: Fake Marks, SS: Secret Sharing, RR: Randomized Response, FHE: Fully Homomorphic Encryption.

## B OPEN SOURCE CODE

We organize open source code of HFedRec *up to February 2024*. Table 8 shows that Python is the mainstream programming language, and existing open source projects have incorporated types of privacy-preserving techniques. TensorFlow, PyTorch, Keras, Rectorch (https://github.com/makgyver/rectorch), and DeepCTR (https://github.com/shenweichen/DeepCTR-Torch) frameworks are utilized to build neural networks. In particular, Rectorch and DeepCTR are developed based on PyTorch and tailor-made for RecSys.

## ACKNOWLEDGMENTS

The authors express gratitude to Professor Qiang Yang for his valuable suggestions and also extend thanks to Dr. Wenyi Zhang and Dr. Siyue Shuai for their partial efforts in enhancing this work.

## REFERENCES


[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*. ACM, New York, NY, USA, 308–318.

[2] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys* 51, 4 (July 2018), Article 79, 35 pages.

[3] Noura Al Ebri, Joonsang Baek, and Chan Yeob Yeun. 2011. Study on secret sharing schemes (SSS) and their applications. In *Proceedings of the 2011 International Conference for Internet Technology and Secured Transactions*. 40–45.

[4] Zareen Alamgir, Farwa K. Khan, and Saira Karim. 2022. Federated recommenders: Methods, challenges and future. *Cluster Computing* 25, 6 (2022), 4075–4096.

[5] Muhammad Ammad-Ud-Din, Elena Ivannikova, Suleiman A. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. 2019. Federated collaborative filtering for privacy-preserving personalized recommendation system. arxiv:1901.09888 (2019).

[6] Alessia Antelmi, Gennaro Cordasco, Mirko Polato, Vittorio Scarano, Carmine Spagnuolo, and Dingqi Yang. 2023. A survey on hypergraph representation learning. *ACM Computing Surveys* 56, 1 (Aug. 2023), Article 24, 38 pages.

[7] Muhammad Asad, Saima Shaukat, Ehsan Javanmardi, Jin Nakazato, and Manabu Tsukada. 2023. A comprehensive survey on privacy-preserving techniques in federated recommendation systems. *Applied Sciences* 13, 10 (2023), 6201.

[8] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47, 2–3 (May 2002), 235–256.

[9] D. Beaver, S. Micali, and P. Rogaway. 1990. The round complexity of secure protocols. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC'90)*. ACM, New York, NY, USA, 503–513.

[10] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. 2012. Foundations of garbled circuits. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS'12)*. ACM, New York, NY, USA, 784–796.

[11] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, Hugh Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*. ACM, New York, NY, USA, 1175–1191.

[12] Longbing Cao. 2016. Non-IID recommender systems: A review and framework of recommendation paradigm shifting. *Engineering* 2, 2 (2016), 212–224.

[13] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2021. Secure federated matrix factorization. *IEEE Intelligent Systems* 36, 5 (2021), 11–20.

[14] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. 2022. Efficient federated matrix factorization against inference attacks. *ACM Transactions on Intelligent Systems and Technology* 13, 4 (June 2022), Article 59, 20 pages.

[15] Moses Charikar, Kevin Chen, and Martin Farach-Colton. 2004. Finding frequent items in data streams. *Theoretical Computer Science* 312, 1 (2004), 3–15.

[16] Chen Chen, Jingfeng Zhang, Anthony K. H. Tung, Mohan Kankanhalli, and Gang Chen. 2020. Robust federated recommendation system. arxiv:2006.08259 (2020).

[17] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2019. Federated meta-learning with fast convergence and efficient communication. arxiv:1802.07876 (2019).

[18] Hao Chen, Kim Laine, and Peter Rindal. 2017. Fast private set intersection from homomorphic encryption. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*. ACM, New York, NY, USA, 1243–1255.

[19] Huiqiang Chen, Tianqing Zhu, Tao Zhang, Wanlei Zhou, and Philip S. Yu. 2023. Privacy and fairness in federated learning: On the perspective of tradeoff. *ACM Computing Surveys* 56, 2 (Sept. 2023), Article 39, 37 pages.

[20] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), Article 67, 39 pages.

[21] Mingyang Chen, Wen Zhang, Zonggang Yuan, Yantao Jia, and Huajun Chen. 2022. FedE: Embedding knowledge graphs in federated setting. In *Proceedings of the 10th International Joint Conference on Knowledge Graphs (IJCKG'21)*. ACM, New York, NY, USA, 80–88.

[22] Wei-Ning Chen, Ayfer Ozgur, and Peter Kairouz. 2022. The Poisson binomial mechanism for unbiased federated learning with secure aggregation. In *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162. 3490–3506.

[23] Yucheng Chen, Chenyuan Feng, and Daquan Feng. 2023. Privacy-preserving hierarchical federated recommendation systems. *IEEE Communications Letters* 27, 5 (2023), 1312–1316.

[24] Yao Chen, Yijie Gui, Hong Lin, Wensheng Gan, and Yongdong Wu. 2022. Federated learning attacks and defenses: A survey. In *Proceedings of the IEEE International Conference on Big Data*. 4256–4265.

[25] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2017. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology—ASIACRYPT 2017*. Springer, Cham, 409–437.

[26] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: Scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*. ACM, New York, NY, USA, 271–280.

[27] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Michel Joseph Lambert, Black Livingston, and Dasarathi Sampath. 2010. The YouTube video recommendation system. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys'10)*. ACM, New York, NY, USA, 293–296.

[28] Aminu Da'u and Naomie Salim. 2020. Recommendation system based on deep learning methods: A systematic review and new directions. *Artificial Intelligence Review* 53, 4 (2020), 2709–2748.

[29] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting telemetry data privately. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. 3574–3583.

[30] Xuanang Ding, Guohui Li, Ling Yuan, Lu Zhang, and Qian Rong. 2023. Efficient federated item similarity model for privacy-preserving recommendation. *Information Processing & Management* 60, 5 (2023), 103470.

[31] Qian Dong, Baisong Liu, Xueyuan Zhang, Jiangcheng Qin, Bingyuan Wang, and Jiangbo Qian. 2022. Ranking-based federated POI recommendation with geographic effect. In *Proceedings of the International Joint Conference on Neural Networks*.

[32] Yongjie Du, Deyun Zhou, Yu Xie, Jiao Shi, and Maoguo Gong. 2021. Federated matrix factorization for privacy-preserving recommender systems. *Applied Soft Computing* 111 (2021), 107700.

[33] Sijing Duan, Deyu Zhang, Yanbo Wang, Lingxiang Li, and Yaoxue Zhang. 2020. JointRec: A deep-learning-based joint cloud video recommendation framework for mobile IoT. *IEEE Internet of Things Journal* 7, 3 (2020), 1655–1666.

[34] Erika Duriakova, Elias Z. Tragos, Barry Smyth, Neil Hurley, Francisco J. Peña, Panagiotis Symeonidis, James Geraci, and Aonghus Lawlor. 2019. PDMFRec: A decentralised matrix factorisation with tunable user-centric privacy. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys'19)*. ACM, New York, NY, USA, 457–461.

[35] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*. Springer, Berlin, Germany, 265–284.

[36] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3–4 (Aug. 2014), 211–407.

[37] Zekeriya Erkin, Thijs Veugen, Tomas Toft, and Reginald L. Lagendijk. 2012. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE Transactions on Information Forensics and Security* 7, 3 (2012), 1053–1066.

[38] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS'14)*. ACM, New York, NY, USA, 1054–1067.

[39] Yahya H. Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and A. Salman Avestimehr. 2023. FairFed: Enabling group fairness in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 7494–7502.

[40] Paul Feldman. 1987. A practical scheme for non-interactive verifiable secret sharing. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science (SFCS'87)*. IEEE, 427–438.

[41] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. 1126–1135.

[42] Adrian Flanagan, Were Oyomno, Alexander Grigorievskiy, Kuan E. Tan, Suleiman A. Khan, and Muhammad Ammad-Ud-Din. 2021. Federated multi-view matrix factorization for personalized recommendations. In *Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, Cham, 324–347.

[43] Tim Ginker and Offer Lieberman. 2021. LSTUR regression theory and the instability of the sample correlation coefficient between financial return indices. *Econometrics Journal* 24, 1 (2021), 58–82.

[44] G. H. Golub and C. Reinsch. 1971. Singular value decomposition and least squares solutions. In *Handbook for Automatic Computation: Volume II: Linear Algebra*. Springer, Berlin, Germany, 134–151.

[45] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129 (2021), 1789–1819.

[46] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2022. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2022), 3549–3568.

[47] Vipul Gupta, Dhruv Choudhary, Peter Tang, Xiaohan Wei, Xing Wang, Yuzhen Huang, Arun Kejariwal, Kannan Ramchandran, and Michael W. Mahoney. 2021. Training recommender systems at scale: Communication-efficient model and data parallelism. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21)*. ACM, New York, NY, USA, 2928–2936.

[48] Yiwen Han, Ding Li, Haotian Qi, Jianji Ren, and Xiaofei Wang. 2019. Federated learning-based computation offloading optimization in edge computing-supported Internet of Things. In *Proceedings of the ACM Turing Celebration Conference—China (ACM TURC'19)*. ACM, New York, NY, USA, Article 25, 5 pages.

[49] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 639–648.

[50] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the Web Conference (WWW'17)*. 173–182.

[51] Seira Hidano and Takao Murakami. 2022. Degree-preserving randomized response for graph neural networks under local differential privacy. arxiv:2202.10209 (2022).

[52] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations*.

[53] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. arxiv:1503.02531 (2015).

[54] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2022. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 44, 9 (Sept. 2022), 5149–5169.

[55] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys* 54, 11s (Sept. 2022), Article 235, 37 pages.

[56] Rui Hu, Yuanxiong Guo, and Yanmin Gong. 2023. Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy. *IEEE Transactions on Mobile Computing*. Early Access, December 14, 2023.

[57] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'08)*. IEEE, 263–272.

[58] Haochen Hua, Yutong Li, Tonghe Wang, Nanqing Dong, Wei Li, and Junwei Cao. 2023. Edge computing with artificial intelligence: A machine learning perspective. *ACM Computing Surveys* 55, 9 (Jan. 2023), 35 pages.

[59] Jiwei Huang, Bowen Ma, Ming Wang, Xiaokang Zhou, Lina Yao, Shoujin Wang, Lianyong Qi, and Ying Chen. 2023. Incentive mechanism design of federated learning for recommendation systems in MEC. *IEEE Transactions on Consumer Electronics*. Published Online, December 13, 2023.

[60] Dzmitry Huba, John Nguyen, Kshitiz Malik, Ruiyu Zhu, Mike Rabbat, Ashkan Yousefpour, Carole-Jean Wu, Hongyuan Zhan, Pavel Ustinov, Harish Srinivas, Kaikai Wang, Anthony Shoumikhin, Jesik Min, and Mani Malek. 2022. Papaya: Practical, private, and scalable federated learning. *Proceedings of Machine Learning and Systems* 4 (2022), 814–832.

[61] Mubashir Imran, Hongzhi Yin, Tong Chen, Quoc Viet Hung Nguyen, Alexander Zhou, and Kai Zheng. 2023. ReFRS: Resource-efficient federated recommender system for dynamic and diversified user preferences. *ACM Transactions on Information Systems* 41, 3 (Feb. 2023), Article 65, 30 pages.

[62] Ahmed Imteaj, Urmish Thakker, Shiqiang Wang, Jian Li, and M. Hadi Amini. 2022. A survey on federated learning for resource-constrained IoT devices. *IEEE Internet of Things Journal* 9, 1 (2022), 1–24.

[63] Md. Ashraful Islam, Mir Mahathir Mohammad, Sarkar Snigdha Sarathi Das, and Mohammed Eunus Ali. 2022. A survey on deep learning based point-of-interest (POI) recommendations. *Neurocomputing* 472 (2022), 306–325.

[64] Jiayun Jiang, Chengte Li, and Shoude Lin. 2019. Towards a more reliable privacy-preserving recommender system. *Information Sciences* 482 (2019), 248–265.

[65] Xueyong Jiang, Baisong Liu, Jiangchen Qin, Yunchong Zhang, and Jiangbo Qian. 2022. FedNCF: Federated neural collaborative filtering for privacy-preserving recommender system. In *Proceedings of the International Joint Conference on Neural Networks*.

[66] Zhifeng Jiang, Wei Wang, and Yang Liu. 2021. FLASHE: Additively symmetric homomorphic encryption for cross-silo federated learning. arxiv:2109.00675 (2021).

[67] Hao Jin, Yang Peng, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2022. Federated reinforcement learning with environment heterogeneity. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, Vol. 151. 18–37.

[68] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. 2019. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *Proceedings of the International Conference on Learning Representations*.

[69] Peter Kairouz, Hugh Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nintin Bhagoji, Kallista Bonawit, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adria Gascon, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecny, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrede Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Ozgur, Rasmus Pagh, Hang Qi,

Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramer, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2021. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning* 14, 1–2 (2021), 1–210.

[70] Saikishore Kalloori and Severin Klingler. 2021. Horizontal cross-silo federated recommender systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*. ACM, New York, NY, USA, 680–684.

[71] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining*. 197–206.

[72] Farwa K. Khan, Adrian Flanagan, Kuan Eeik Tan, Zareen Alamgir, and Muhammad Ammad-Ud Din. 2021. A payload optimization method for federated recommender systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*. ACM, New York, NY, USA, 432–442.

[73] Hiroaki Kikuchi, Yoshiki Aoki, Masayuki Terada, Kazuki Ishii, and Kimihiko Sekino. 2012. Accuracy of privacy-preserving collaborative filtering based on quasi-homomorphic similarity. In *Proceedings of the 2012 9th International Conference on Ubiquitous Intelligence and Computing and the 9th International Conference on Autonomic and Trusted Computing*. 555–562.

[74] Hiroaki Kikuchi and Anna Mochizuki. 2012. Privacy-preserving collaborative filtering using randomized response. In *Proceedings of the 2012 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'12)*. IEEE, 671–676.

[75] Sungwook Kim, Jinsu Kim, Dongyoung Koo, Yuna Kim, Hyunsoo Yoon, and Junbum Shin. 2016. Efficient privacy-preserving matrix factorization via fully homomorphic encryption: Extended abstract. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS'16)*. ACM, New York, NY, USA, 617–628.

[76] Vladimir Kolesnikov and Thomas Schneider. 2008. Improved garbled circuit: Free XOR gates and applications. In *Proceedings of the 35th International Colloquium on Automata, Languages, and Programming, Part II (ICALP'08)*. 486–498.

[77] Xiangjie Kong, Yuhan Wu, Hui Wang, and Feng Xia. 2022. Edge computing for Internet of Everything: A survey. *IEEE Internet of Things Journal* 9, 23 (2022), 23472–23485.

[78] Emmanouil Krasanakis, Symeon Papadopoulos, and Ioannis Kompatsiaris. 2022. p2pGNN: A decentralized graph neural network for node classification in peer-to-peer networks. *IEEE Access* 10 (2022), 34755–34765.

[79] Daliang Li and Junpu Wang. 2019. FedMD: Heterogenous federated learning via model distillation. arxiv:1910.03581 (2019).

[80] Li Li, Fan Lin, Jianbing Xiahou, Yuanguo Lin, Pengcheng Wu, and Yong Liu. 2022. Federated low-rank tensor projections for sequential recommendation. *Knowledge-Based Systems* 255 (2022), 109483.

[81] Quan Li, Xiguang Wei, Huanbin Lin, Yang Liu, Tianjian Chen, and Xiaojuan Ma. 2022. Inspecting the running process of horizontal federated learning via visual analytics. *IEEE Transactions on Visualization and Computer Graphics* 28, 12 (2022), 4085–4100.

[82] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning*, Vol. 139. 6357–6368.

[83] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* 2 (2020), 429–450.

[84] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2020. Fair resource allocation in federated learning. In *Proceedings of the International Conference on Learning Representations*.

[85] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. 2021. FedBN: Federated learning on non-IID features via local batch normalization. In *Proceedings of the International Conference on Learning Representations*.

[86] Youhuizi Li, Haitao Yu, Yan Zeng, and Qianqian Pan. 2023. HFSA: A semi-asynchronous hierarchical federated recommendation system in smart city. *IEEE Internet of Things Journal* 10, 21 (2023), 18808–18820.

[87] Zheng Li, Muhammad Bilal, Xiaolong Xu, Jielin Jiang, and Yan Cui. 2023. Federated learning-based cross-enterprise recommendation with graph neural networks. *IEEE Transactions on Industrial Informatics* 19, 1 (2023), 673–682.

[88] Zexi Li, Qunwei Li, Yi Zhou, Wenliang Zhong, Guannan Zhang, and Chao Wu. 2023. Edge-cloud collaborative learning with federated and centralized features. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'23)*. ACM, New York, NY, USA, 1949–1953.

[89] Zhitao Li, Zhaohao Lin, Feng Liang, Weike Pan, Qiang Yang, and Zhong Ming. 2024. Decentralized federated recommendation with privacy-aware structured client-level graph. *ACM Transactions on Intelligent Systems and Technology*. Accepted, January 2024.

[90] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2022. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems* 33, 12 (2022), 6999–7019.

[91] Feng Liang, Weike Pan, and Zhong Ming. 2021. FedRec++: Lossless federated recommendation with explicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4224–4231.
[92] Guanyu Lin, Feng Liang, Weike Pan, and Zhong Ming. 2021. FedRec: Federated recommendation with explicit feedback. *IEEE Intelligent Systems* 36, 5 (2021), 21–30.
[93] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Dongxiao Yu, Jun Ma, Maarten de Rijke, and Xiuzhen Cheng. 2020. Meta matrix factorization for federated rating predictions. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 981–990.
[94] Zhaohao Lin, Weike Pan, Qiang Yang, and Zhong Ming. 2022. A generic federated recommendation framework via fake marks and secret sharing. *ACM Transactions on Information Systems* 41, 2 (Dec. 2022), Article 40, 37 pages.
[95] Yehuda Lindell and Benny Pinkas. 2015. An efficient protocol for secure two-party computation in the presence of malicious adversaries. *Journal of Cryptology* 28, 2 (April 2015), 312–350.
[96] Linfeng Liu, Yaoze Zhou, and Jia Xu. 2023. A cloud-edge-end collaboration framework for cruising route recommendation of vacant taxis. *IEEE Transactions on Mobile Computing*. Published Online, July 13, 2023.
[97] Ruixuan Liu, Yang Cao, Yanlin Wang, Lingjuan Lyu, Yun Chen, and Hong Chen. 2023. PrivateRec: Differentially private model training and online serving for federated news recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'23)*. ACM, New York, NY, USA, 4539–4548.
[98] Shuchang Liu, Yingqiang Ge, Shuyuan Xu, Yongfeng Zhang, and Amelie Marian. 2022. Fairness-aware federated matrix factorization. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys'22)*. ACM, New York, NY, USA, 168–178.
[99] Liu Yang, Ben Tan, Vincent W. Zheng, Kai Chen, and Qiang Yang. 2020. Federated recommendation systems. In *Federated Learning: Privacy and Incentive*. Springer, Cham, 225–239.
[100] Zhi Liu, Yang Chen, Feng Xia, Jixin Bian, Bing Zhu, Guojiang Shen, and Xiangjie Kong. 2023. TAP: Traffic accident profiling via multi-task spatio-temporal graph representation learning. *ACM Transactions on Knowledge Discovery from Data* 17, 4 (Feb. 2023), Article 56, 25 pages.
[101] Zengyan Liu, Linqi Song, and Christina Fragouli. 2022. Federated multi-armed bandits with vector rewards for aspect-based recommendations. In *Proceedings of the IEEE Global Communications Conference*. 1079–1084.
[102] Zhiwei Liu, Liangwei Yang, Ziwei Fan, Hao Peng, and Philip S. Yu. 2022. Federated social recommendation with graph neural network. *ACM Transactions on Intelligent Systems and Technology* 13, 4 (Aug. 2022), Article 55, 24 pages.
[103] Linze Luo and Baisong Liu. 2022. Dual-contrastive for federated social recommendation. In *Proceedings of the International Joint Conference on Neural Networks*.
[104] Sichun Luo, Yuanzhang Xiao, Yang Liu, Congduan Li, and Linqi Song. 2022. Towards communication efficient and fair federated personalized sequential recommendation. In *Proceedings of the International Conference on Information Communication and Signal Processing*.
[105] Sichun Luo, Yuanzhang Xiao, and Linqi Song. 2022. Personalized federated recommendation via joint representation learning, user clustering, and model adaptation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM'22)*. ACM, New York, NY, USA, 4289–4293.
[106] Sichun Luo, Yuanzhang Xiao, Xinyi Zhang, Yang Liu, Wenbo Ding, and Linqi Song. 2023. PerFedRec++: Enhancing personalized federated recommendation with self-supervised pre-training. arxiv:2305.06622 (2023).
[107] Chuang Ma, Xin Ren, Guangxia Xu, and Bo He. 2023. FedGR: Federated graph neural network for recommendation systems. *Axioms* 12, 2 (2023), 170.
[108] Xiaodong Ma, Jia Zhu, Zhihao Lin, Shanxuan Chen, and Yangjie Qin. 2022. A state-of-the-art survey on solving non-IID data in federated learning. *Future Generation Computer Systems* 135 (2022), 244–258.
[109] Kiwan Maeng, Haiyu Lu, Luca Melis, John Nguyen, Mike Rabbat, and Carole-Jean Wu. 2022. Towards fair federated recommendation learning: Characterizing the inter-dependence of system and data heterogeneity. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys'22)*. ACM, New York, NY, USA, 156–167.
[110] Peihua Mai and Yan Pang. 2024. Privacy-preserving multiview matrix factorization for recommender systems. *IEEE Transactions on Artificial Intelligence* 5, 1 (2024), 267–277.
[111] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Vol. 54. 1273–1282.
[112] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning differentially private recurrent language models. In *Proceedings of the International Conference on Learning Representations*.
[113] Frank McSherry and Ilya Mironov. 2009. Differentially private recommender systems: Building privacy into the Netflix Prize contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. ACM, New York, NY, USA, 627–636.

[114] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys* 54, 6 (2021), Article 115, 35 pages.

[115] Lorenzo Minto, Moritz Haller, Benjamin Livshits, and Hamed Haddadi. 2021. Stronger privacy for federated collaborative filtering with implicit feedback. In *Proceedings of the 15th ACM Conference on Recommender Systems*. ACM, New York, NY, USA, 342–350.

[116] Ilya Mironov. 2017. Rényi differential privacy. In *Proceedings of the IEEE 30th Computer Security Foundations Symposium (CSF'17)*. IEEE, 263–275.

[117] Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. 2020. FedFast: Going beyond average for faster training of federated recommender systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'20)*. ACM, New York, NY, USA, 1234–1242.

[118] Moni Naor and Benny Pinkas. 1999. Oblivious transfer and polynomial evaluation. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC'99)*. ACM, New York, NY, USA, 245–254.

[119] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy (SP'08)*. IEEE, 111–125.

[120] Kartik Nayak, Xiao Shaun Wang, Stratis Ioannidis, Udi Weinsberg, Nina Taft, and Elaine Shi. 2015. GraphSC: Parallel secure computation made easy. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy (SP'15)*. IEEE, 377–394.

[121] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. arxiv:1803.02999 (2018).

[122] Valeria Nikolaenko, Stratis Ioannidis, Udi Weinsberg, Marc Joye, Nina Taft, and Dan Boneh. 2013. Privacy-preserving matrix factorization. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security (CCS'13)*. ACM, New York, NY, USA, 801–812.

[123] Lin Ning, Karan Singhal, Ellie X. Zhou, and Sushant Prakash. 2021. Learning federated representations and recommendations with limited negatives. arxiv:2108.07931 (2021).

[124] Pascal Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT'99)*. 223–238.

[125] Shubham Pateria, Budhitama Subagdja, Ah-Hwee Tan, and Chai Quek. 2021. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys* 54, 5 (June 2021), Article 109, 35 pages.

[126] Hao Peng, Haoran Li, Yangqiu Song, Vincent Zheng, and Jianxin Li. 2021. Differentially private federated knowledge graphs embedding. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. ACM, New York, NY, USA, 1416–1425.

[127] Zhuoyi Peng, Yi Yang, Liu Yang, and Kai Chen. 2022. Federated meta embedding concept stock recommendation. *IEEE Transactions on Big Data*. Early Access, October 14, 2022.

[128] Vasileios Perifanis, George Drosatos, Giorgos Stamatelatos, and Pavlos S. Efraimidis. 2023. FedPOIRec: Privacy-preserving federated POI recommendation with social influence. *Information Sciences* 623, C (April 2023), 767–790.

[129] Vasileios Perifanis and Pavlos S. Efraimidis. 2022. Federated neural collaborative filtering. *Knowledge-Based Systems* 242 (2022), 108441.

[130] H. Polat and Wenliang Du. 2003. Privacy-preserving collaborative filtering using randomized perturbation techniques. In *Proceedings of the IEEE International Conference on Data Mining*. IEEE, 625–628.

[131] Mirko Polato. 2021. Federated variational autoencoder for collaborative filtering. In *Proceedings of the International Joint Conference on Neural Networks*.

[132] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S. Iyengar. 2018. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys* 51, 5 (2018), 1–36.

[133] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-preserving news recommendation model learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 1423–1432.

[134] Zhen Qin, Shuiguang Deng, Mingyu Zhao, and Xueqiang Yan. 2023. FedAPEN: Personalized cross-silo federated learning with adaptability to statistical heterogeneity. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'23)*. ACM, New York, NY, USA, 1954–1964.

[135] Liang Qu, Ningzhi Tang, Ruiqi Zheng, Quoc Viet Hung Nguyen, Zi Huang, Yuhui Shi, and Hongzhi Yin. 2023. Semi-decentralized federated ego graph learning for recommendation. In *Proceedings of the ACM Web Conference 2023 (WWW'23)*. ACM, New York, NY, USA, 339–348.

[136] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *ACM Computing Surveys* 51, 4 (2018), Article 66, 36 pages.

[137] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI'09)*. 452–461.

[138] Dazhong Rong, Qinming He, and Jianhai Chen. 2022. Poisoning deep learning based recommender model in federated learning scenarios. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*.

[139] Dazhong Rong, Shuai Ye, Ruoyan Zhao, Hon Ning Yuen, Jianhai Chen, and Qinming He. 2022. FedRecAttack: Model poisoning attack to federated recommendation. In *Proceedings of the IEEE 38th International Conference on Data Engineering*. IEEE, 2643–2655.

[140] Haidong Rong, Yangzihao Wang, Feihu Zhou, Junjie Zhai, Haiyang Wu, Rui Lan, Fan Li, Han Zhang, Yuekui Yang, Zhenyu Guo, and Di Wang. 2020. Distributed equivalent substitution training for large-scale recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 911–920.

[141] Luca Rossi, Matthew Williams, Christoph Stich, and Mirco Musolesi. 2015. Privacy and the city: User identification and location semantics in location-based social networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 9. 387–396.

[142] Sefutsu Ryu and Shinya Takamaeda-Yamazaki. 2022. Model-based federated reinforcement distillation. In *Proceedings of the IEEE Global Communications Conference*. 1109–1114.

[143] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'07)*. 1257–1264.

[144] Adi Shamir. 1979. How to share a secret. *Communications of the ACM* 22, 11 (Nov. 1979), 612–613.

[145] Huajie Shao, Zhisheng Xiao, Shuochao Yao, Dachun Sun, Aston Zhang, Shengzhong Liu, Tianshi Wang, Jinyang Li, and Tarek Abdelzaher. 2022. ControlVAE: Tuning, analytical properties, and performance analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 12 (2022), 9285–9297.

[146] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Kun Kuang, Fei Wu, and Chao Wu. 2020. Federated mutual learning. arxiv:2006.16765 (2020).

[147] Chengshuai Shi and Cong Shen. 2021. Federated multi-armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 9603–9611.

[148] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. 2021. Masked label prediction: Unified message passing model for semi-supervised classification. In *Proceedings of the International Joint Conferences on Artificial Intelligence*.

[149] Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. ACM, New York, NY, USA, 650–658.

[150] Jinhyun So, Başak Güler, and A. Salman Avestimehr. 2021. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning. *IEEE Journal on Selected Areas in Information Theory* 2, 1 (2021), 479–489.

[151] Statista.com. 2023. Number of internet and social media users worldwide as of April 2023. Retrieved July 16, 2023 from https://www.statista.com/statistics/617136/digital-population-worldwide

[152] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM'19)*. ACM, New York, NY, USA, 1441–1450.

[153] Feng Sun, Zhenjiang Zhang, Sherali Zeadally, Guangjie Han, and Shiyuan Tong. 2022. Edge computing-enabled internet of vehicles: Towards federated learning empowered scheduling. *IEEE Transactions on Vehicular Technology* 71, 9 (2022), 10088–10103.

[154] Zehua Sun, Yonghui Xu, Yong Liu, Wei He, Lanju Kong, Fangzhao Wu, Yali Jiang, and Lizhen Cui. 2024. A survey on federated recommendation systems. *IEEE Transactions on Neural Networks and Learning Systems*. Early Access, February 7, 2024.

[155] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2023. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems* 34, 12 (2023), 9587–9603.

[156] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2022. FedProto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8432–8440.

[157] Jiaxi Tang and Ke Wang. 2018. Personalized top-$N$ sequential recommendation via convolutional sequence embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM'18)*. ACM, New York, NY, USA, 565–573.

[158] Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. 2022. SplitFed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8485–8493.

[159] Antonela Tommasel, Juan Manuel Rodriguez, and Daniela Godoy. 2021. I want to break free! Recommending friends from outside the echo chamber. In *Proceedings of the 15th ACM Conference on Recommender Systems*. ACM, New York, NY, USA, 23–33.

[160] Omar Abdel Wahab, Gaith Rjoub, Jamal Bentahar, and Robin Cohen. 2022. Federated against the cold: A trust-based federated learning approach to counter the cold start problem in recommendation systems. *Information Sciences* 601 (2022), 189–206.

[161] Jingdong Wang, Ting Zhang, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. 2018. A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 4 (2018), 769–790.

[162] Le Wang, Zijun Huang, Qingqi Pei, and Shen Wang. 2020. Federated CF: Privacy-preserving collaborative filtering cross multiple datasets. In *Proceedings of the IEEE International Conference on Communications*.

[163] Qinyong Wang, Hongzhi Yin, Tong Chen, Junliang Yu, Alexander Zhou, and Xiangliang Zhang. 2021. Fast-adapting and privacy-preserving federated recommender system. *VLDB Journal* 31, 5 (Oct. 2021), 877–896.

[164] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. 2021. A survey on session-based recommender systems. *ACM Computing Surveys* 54, 7 (July 2021), Article 154, 38 pages.

[165] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: Challenges, progress and prospects. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 6332–6338.

[166] Shanfeng Wang, Hao Tao, Jianzhao Li, Xinyuan Ji, Yuan Gao, and Maoguo Gong. 2024. Towards fair and personalized federated recommendation. *Pattern Recognition* 149 (2024), 110234.

[167] Xiaojie Wang, Jiameng Li, Zhaolong Ning, Qingyang Song, Lei Guo, Song Guo, and Mohammad S. Obaidat. 2023. Wireless powered mobile edge computing networks: A survey. *ACM Computing Surveys* 55, 13s (July 2023), Article 263, 37 pages.

[168] Yujia Wang, Lu Lin, and Jinghui Chen. 2022. Communication-efficient adaptive federated learning. In *Proceedings of the 39th International Conference on Machine Learning*, Vol. 162. 22802–22838.

[169] Stanley L. Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association* 60, 309 (1965), 63–69.

[170] Kang Wei, Jun Li, Chuan Ma, Ming Ding, Wen Chen, Jun Wu, Meixia Tao, and H. Vincent Poor. 2023. Personalized federated learning with differential privacy and convergence guarantee. *IEEE Transactions on Information Forensics and Security* 18 (2023), 4488–4503.

[171] Shanming Wei, Shunmei Meng, Qianmu Li, Xiaokang Zhou, Lianyong Qi, and Xiaolong Xu. 2023. Edge-enabled federated sequential recommendation with knowledge-aware transformer. *Future Generation Computer Systems* 148 (2023), 610–622.

[172] Udi Weinsberg, Smriti Bhagat, Stratis Ioannidis, and Nina Taft. 2012. BlurMe: Inferring and obfuscating user gender based on ratings. In *Proceedings of the 6th ACM Conference on Recommender Systems (RecSys'12)*. ACM, New York, NY, USA, 195–202.

[173] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. Communication-efficient federated learning via knowledge distillation. *Nature Communications* 13, 1 (2022), 2032.

[174] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Tao Qi, Yongfeng Huang, and Xing Xie. 2022. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications* 13, 1 (2022), 3091.

[175] Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2022. FedAttack: Effective and covert poisoning attack on federated recommendation via hard sampling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, 4164–4172.

[176] Danye Wu, Miao Pan, Zhiwei Xu, Yujun Zhang, and Zhu Han. 2020. Towards efficient secure aggregation for model update in federated learning. In *Proceedings of the IEEE Global Communications Conference*.

[177] Jinze Wu, Qi Liu, Zhenya Huang, Yuting Ning, Hao Wang, Enhong Chen, Jinfeng Yi, and Bowen Zhou. 2021. Hierarchical personalized federated learning for user modeling. In *Proceedings of the Web Conference (WWW'21)*. ACM, New York, NY, USA, 957–968.

[178] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: A survey. *ACM Computing Surveys* 55, 5 (Dec. 2022), Article 97, 37 pages.

[179] Wei Wu, Jian Liu, Huimei Wang, Jialu Hao, and Ming Xian. 2021. Secure and efficient outsourced *k*-means clustering using fully homomorphic encryption with ciphertext packing technique. *IEEE Transactions on Knowledge and Data Engineering* 33, 10 (2021), 3424–3437.

[180] Zhaomin Wu, Qinbin Li, and Bingsheng He. 2022. Practical vertical federated learning with unsupervised representation learning. *IEEE Transactions on Big Data*. Published Online, August 13, 2022.

[181] Feng Xia, Lei Wang, Tao Tang, Xin Chen, Xiangjie Kong, Giles Oatley, and Irwin King. 2023. CenGCN: Centralized convolutional networks with vertex imbalance for scale-free graphs. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2023), 4555–4569.

[182] Guiliang Xiao, Mingjun Xiao, Guoju Gao, Sheng Zhang, Hui Zhao, and Xiang Zou. 2020. Incentive mechanism design for federated learning: A two-stage Stackelberg game approach. In *Proceedings of the IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS'20)*. IEEE, 148–155.

[183] Bo Yan, Yang Cao, Haoyu Wang, Wenchuan Yang, Junping Du, and Chuan Shi. 2024. Federated heterogeneous graph neural network for privacy-preserving recommendation. In *Proceedings of the Web Conference*.

[184] Dingqi Yang, Daqing Zhang, Bingqing Qu, and Philippe Cudré-Mauroux. 2016. PrivCheck: Privacy-preserving check-in data publishing for personalized location based services. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'16)*. ACM, New York, NY, USA, 545–556.

[185] Enyue Yang, Yunfeng Huang, Feng Liang, Weike Pan, and Zhong Ming. 2021. FCMF: Federated collective matrix factorization for heterogeneous collaborative filtering. *Knowledge-Based Systems* 220 (2021), 106946.

[186] Kang Yang, Xiao Wang, and Jiang Zhang. 2020. More efficient MPC from improved triple generation and authenticated garbling. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY, USA, 1627–1646.

[187] Lei Yang, Jiaming Huang, Wanyu Lin, and Jiannong Cao. 2023. Personalized federated learning on non-IID data via group-based meta-learning. *ACM Transactions on Knowledge Discovery from Data* 17, 4 (2023), Article 49, 20 pages.

[188] Liu Yang, Junxue Zhang, Di Chai, Leye Wang, Kun Guo, Kai Chen, and Qiang Yang. 2022. Practical and secure federated recommendation with personalized mask. In *Proceedings of the International Workshop on Trustworthy Federated Learning*. 33–45.

[189] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2 (Jan. 2019), Article 12, 19 pages.

[190] Andrew C. Yao. 1982. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS'82)*. IEEE, 160–164.

[191] Mang Ye, Xiuwen Fang, Bo Du, Pong C. Yuen, and Dacheng Tao. 2023. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys* 56, 3 (Oct. 2023), Article 79, 44 pages.

[192] Ziming Ye, Xiao Zhang, Xu Chen, Hui Xiong, and Dongxiao Yu. 2023. Adaptive clustering based personalized federated learning framework for next POI recommendation with location noise. *IEEE Transactions on Knowledge and Data Engineering*. Published Online, September 6, 2023.

[193] Jingwei Yi, Fangzhao Wu, Chuhan Wu, Ruixuan Liu, Guangzhong Sun, and Xing Xie. 2021. Efficient-FedRec: Efficient federated learning framework for privacy-preserving news recommendation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2814–2824.

[194] Hongzhi Yin, Qinyong Wang, Kai Zheng, Zhixu Li, and Xiaofang Zhou. 2022. Overcoming data sparsity in group recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 7 (2022), 3447–3460.

[195] Yuyu Yin, Youhuizi Li, Honghao Gao, Tingting Liang, and Qianqian Pan. 2023. FGC: GCN-based federated learning approach for trust industrial service recommendation. *IEEE Transactions on Industrial Informatics* 19, 3 (2023), 3240–3250.

[196] Senci Ying. 2020. Shared MF: A privacy-preserving recommendation system. arxiv:2008.07759 (2020).

[197] Sanshi Lei Yu, Qi Liu, Fei Wang, Yang Yu, and Enhong Chen. 2023. Federated news recommendation with fine-grained interpolation and dynamic clustering. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM'23)*. ACM, New York, NY, USA, 3073–3082.

[198] Yang Yu, Qi Liu, Likang Wu, Runlong Yu, Sanshi Lei Yu, and Zaixi Zhang. 2023. Untargeted attack against federated recommendation systems via poisonous item embeddings and the defense. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4854–4863.

[199] Wei Yuan, Quoc Viet Hung Nguyen, Tieke He, Liang Chen, and Hongzhi Yin. 2023. Manipulating federated recommender systems: Poisoning with synthetic users and its countermeasures. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'23)*. ACM, New York, NY, USA, 1690–1699.

[200] Wei Yuan, Liang Qu, Lizhen Cui, Yongxin Tong, Xiaofang Zhou, and Hongzhi Yin. 2024. HeteFedRec: Federated recommender systems with model heterogeneity. In *Proceedings of the IEEE 40th International Conference on Data Engineering*.

[201] Wei Yuan, Chaoqun Yang, Quoc Viet Hung Nguyen, Lizhen Cui, Tieke He, and Hongzhi Yin. 2023. Interaction-level membership inference attack against federated recommender systems. In *Proceedings of the Web Conference (WWW'23)*. ACM, New York, NY, USA, 1053–1062.

[202] Samee Zahur, Mike Rosulek, and David Evans. 2015. Two halves make a whole. In *Advances in Cryptology—EUROCRYPT 2015*. Springer, Berlin, Germany, 220–250.

[203] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. 2020. BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning. In *Proceedings of the 2020 USENIX Annual Technical Conference*.

[204] Chunxu Zhang, Guodong Long, Tianyi Zhou, Peng Yan, Zijjian Zhang, and Bo Yang. 2023. Graph-guided personalization for federated recommendation. arxiv:2305.07866 (2023).

[205] Chunxu Zhang, Guodong Long, Tianyi Zhou, Peng Yan, Zijian Zhang, Chengqi Zhang, and Bo Yang. 2023. Dual personalization on federated recommendation. In *Proceedings of the International Joint Conferences on Artificial Intelligence*.

[206] Chunxu Zhang, Guodong Long, Tianyi Zhou, Xiangyu Zhao, Zijian Zhang, and Bo Yang. 2023. IFedRec: Item-guided federated aggregation for cold-start. arxiv:2305.12650 (2023).

[207] Guijuan Zhang, Yang Liu, and Xiaoning Jin. 2020. A survey of autoencoder-based recommender systems. *Frontiers of Computer Science* 14, 2 (2020), 430–450.

[208] Honglei Zhang, Fangyuan Luo, Jun Wu, Xiangnan He, and Yidong Li. 2023. LightFR: Lightweight federated recommendation with privacy-preserving matrix factorization. *ACM Transactions on Information Systems* 41, 4 (2023), Article 90, 28 pages.

[209] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. 2019. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *Proceedings of the International Conference on Learning Representations*.

[210] Kaiqing Zhang, Tao Sun, Yunzhe Tao, Sahika Genc, Sunil Mallya, and Tamer Basar. 2020. Robust multi-agent reinforcement learning with model uncertainty. *Advances in Neural Information Processing Systems* 33 (2020), 10571–10583.

[211] Lu Zhang, Guohui Li, Ling Yuan, Xuanang Ding, and Qian Rong. 2024. HN3S: A federated autoencoder framework for collaborative filtering via hybrid negative sampling and secret sharing. *Information Processing & Management* 61, 2 (2024), 103580.

[212] Mengmei Zhang, Xiao Wang, Meiqi Zhu, Chuan Shi, Zhiqiang Zhang, and Jun Zhou. 2022. Robust heterogeneous graph neural networks against adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4363–4370.

[213] Meng Zhang, Ermin Wei, and Randall Berry. 2021. Faithful edge federated learning: Scalability and privacy. *IEEE Journal on Selected Areas in Communications* 39, 12 (2021), 3790–3804.

[214] Rongyu Zhang, Yun Chen, Chenrui Wu, and Fangxin Wang. 2023. Cluster-driven GNN-based federated recommendation with biased message dropout. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. 594–599.

[215] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys* 52, 1 (Feb. 2019), Article 5, 38 pages.

[216] Shijie Zhang, Hongzhi Yin, Tong Chen, Zi Huang, Quoc Viet Hung Nguyen, and Lizhen Cui. 2022. PipAttack: Poisoning federated recommender systems for manipulating item promotion. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM'22)*. ACM, New York, NY, USA, 1415–1423.

[217] Xiao Zhang, Ziming Ye, Jianfeng Lu, Fuzhen Zhuang, Yanwei Zheng, and Dongxiao Yu. 2023. Fine-grained preference-aware personalized federated POI recommendation with data sparsity. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'23)*. ACM, New York, NY, USA, 413–422.

[218] Xu Zhang, Felix X. Yu, Sanjiv Kumar, and Shih-Fu Chang. 2017. Learning spread-out local feature descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*. 4595–4603.

[219] Yan Zhang, Guojiang Shen, Xiao Han, Wei Wang, and Xiangjie Kong. 2023. Spatio-temporal digraph convolutional network-based taxi pickup location recommendation. *IEEE Transactions on Industrial Informatics* 19, 1 (2023), 394–403.

[220] Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu-An Tan. 2019. Secure multi-party computation: Theory, practice and applications. *Information Sciences* 476 (2019), 357–372.

[221] Shuai Zhao, Roshani Bharati, Cristian Borcea, and Yi Chen. 2020. Privacy-aware federated learning for page recommendation. In *Proceedings of the IEEE International Conference on Big Data*. 1071–1080.

[222] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, Yushuo Chen, Lanling Xu, Gaowei Zhang, Zhen Tian, Changxin Tian, Shanlei Mu, Xinyan Fan, Xu Chen, and Ji-Rong Wen. 2022. RecBole 2.0: Towards a more up-to-date recommendation library. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM'22)*. ACM, New York, NY, USA, 4722–4726.

[223] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM'21)*. ACM, New York, NY, USA, 4653–4664.

[224] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-IID data. arxiv:1806.00582 (2018).

[225] Chong Zheng, Shengheng Liu, Yongming Huang, Wei Zhang, and Luxi Yang. 2022. Unsupervised recurrent federated learning for edge popularity prediction in privacy-preserving mobile-edge computing networks. *IEEE Internet of Things Journal* 9, 23 (2022), 24328–24345.

[226] Xiaoyao Zheng, Manping Guan, Xianmin Jia, Liping Sun, and Yonglong Luo. 2023. Federated matrix factorization recommendation based on secret sharing for privacy preserving. *IEEE Transactions on Computational Social Systems.* Early Access, October 16, 2023.

[227] Xiaolin Zheng, Zhongyu Wang, Chaochao Chen, Jiashu Qian, and Yao Yang. 2023. Decentralized graph neural network for privacy-preserving recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM'23).* ACM, New York, NY, USA, 3494–3504.

[228] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. 2021. Federated learning on non-IID data: A survey. *Neurocomputing* 465 (2021), 371–390.

[229] Jieming Zhu, Quanyu Dai, Liangcai Su, Rong Ma, Jinyang Liu, Guohao Cai, Xi Xiao, and Rui Zhang. 2022. BARS: Towards open benchmarking for recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'22).* ACM, New York, NY, USA, 2912–2923.

[230] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open benchmarking for click-through rate prediction. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM'21).* ACM, New York, NY, USA, 2759–2769.

[231] Zhitao Zhu, Shijing Si, Jianzong Wang, and Jing Xiao. 2022. Cali3F: Calibrated fast fair federated recommendation system. In *Proceedings of the International Joint Conference on Neural Networks.*

Received 15 June 2022; revised 25 February 2024; accepted 31 March 2024